

**DEPTH RECOVERY WITH RECTIFICATION
USING SINGLE-LENS PRISM BASED
STEREOVISION SYSTEM**

WANG DAOLEI

NATIONAL UNIVERSITY OF SINGAPORE

2012

**DEPTH RECOVERY WITH RECTIFICATION
USING SINGLE-LENS PRISM BASED
STEREOVISION SYSTEM**

WANG DAOLEI

(B.S., ZHEJIANG SCI-TECH UNIVERSITY)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

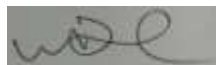
2012

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety.

I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



Wang Daolei

16 August, 2012

ACKNOWLEDGMENTS

I wish to express my gratitude and appreciation to my supervisor, A/Prof. Kah Bin LIM for his instructive guidance and constant personal encouragement during every stage of my Ph.D. study. I gratefully acknowledge the financial support provided by the National University of Singapore (NUS) and China Scholarship Council (CSC) that make it possible for me to finish this study.

I appreciate Dr. Xiao Yong, for his excellent early contribution initiation on single-lens stereovision using a bi-prism (2F-filter).

My gratitude also goes to Mr. Yee, Mrs. Ooi, Ms. Tshin, and Miss Hamidah for their help on facility support in the laboratory so that my research could be completed smoothly.

It is also a true pleasure for me to meet many nice and wise colleagues in the Control and Mechatronics Laboratory, who made the past four years exciting and the experience worthwhile. I am sincerely grateful for the friendship and companionship from Zhang Meijun, Wang Qing, Wu Jiayun, Kee Wei Loon, and Bai Yading, etc.

Finally, I would like to thank my parents, and sisters for their constant love and endless support through my student life. My gratefulness and appreciation cannot be expressed in words.

TABLE OF CONTENTS

DECLARATION	I
ACKNOWLEDGMENTS	II
TABLE OF CONTENTS	III
SUMMARY	VI
LIST OF TABLES	VIII
LIST OF FIGURES	IX
LIST OF ABBREVIATIONS.....	XIII
Chapter 1 Introduction	1
1.1 Background	1
1.2 Problem descriptions.....	2
1.3 Motivation.....	5
1.4 Scope of study and objectives	6
1.5 Outline of the thesis	7
Chapter 2 Literature review	9
2.1 Stereovision systems	9
2.2 Camera calibration	14
2.3 Epipolar geometry constraints	15
2.4 Review of rectification algorithms.....	18
2.5 Stereo correspondence algorithms	20
2.6 Stereo 3-D reconstruction	31
2.7 Summary	32
Chapter 3 Rectification of single-lens binocular stereovision system	33
3.1 The background of stereo vision rectification.....	35
3.2 Rectification of single-lens binocular stereovision system using geometrical approach.....	40
3.2.1 Computation of the virtual cameras' projection matrix	41

3.2.2 Rectification Algorithm	55
3.3 Experimental results and discussion	57
3.4 Summary	65
Chapter 4 Rectification of single-lens trinocular and multi-ocular stereovision system	66
4.1 A geometry-based approach for three-view image rectification	66
4.1.1 Generation of three virtual cameras	67
4.1.2 Determination of the virtual cameras' projection matrix by geometrical analysis of ray sketching	69
4.1.3 Rectification Algorithm	84
4.2 The multi-ocular stereo vision rectification	85
4.3 Experimental results and discussion	89
4.4 Summary	96
Chapter 5 Segment-based stereo matching using cooperative optimization: image segmentation and initial disparity map acquisition.....	98
5.1 Image segmentation	99
5.1.1 Mean-shift method	100
5.1.2 Application of mean-shift method	102
5.2 Initial disparity map acquisition.....	104
5.2.1 Biologically inspired aggregation	104
5.2.2 Initial disparity map estimation algorithm.....	106
5.3 Experimental results and discussion	109
5.3.1 Experimental procedure	110
5.3.2 Experimentation results.....	110
5.3.3 Analysis of results	112
5.4 Summary	113
Chapter 6 Segment-based stereo matching using cooperative optimization: disparity plane estimation and cooperative optimization for energy function.....	115
6.1 Disparity plane estimation	115
6.1.1 Plane fitting	116
6.1.2 Outlier filtering	118
6.1.3 Merging of neighboring disparity planes	122

6.1.4 Experiment	126
6.2 Cooperative optimization of energy function	128
6.2.1 Cooperative optimization algorithm	128
6.2.2 The formulation of energy function	130
6.2.3 Experiment	132
6.3 Summary	137
Chapter 7 Multi-view stereo matching and depth recovery	138
7.1 Multiple views stereo matching	138
7.1.1 Applying the local method to obtain multi-view stereo disparity	140
7.1.2 Applying the global method to obtain multi-view disparity map	142
7.2 Depth recovery	149
7.2.1 Triangulation to general stereo pairs	149
7.2.3 Triangulation to rectified stereo pairs	150
7.3 Experimental results	153
7.3.1 Multi-view stereo matching algorithm results and discussion	153
7.3.2 Depth recovery results and discussion	157
7.4 Summary	162
Chapter 8 Conclusions and future works	163
8.1 Summary and contributions of the thesis	163
8.2 Limitations and Future works	166
Bibliography	168
Appendices	180
List of publications	194

SUMMARY

This thesis aims to study the depth recovery of a 3D scene using a single-lens stereovision system with prism (filter). An image captured by this system (*image acquisition*) is split into multiple different sub-images on the camera image plane. They are assumed to have been captured simultaneously by a group of virtual cameras which are generated by the prism. A point in the scene would appear in different locations in each of the image planes, and the differences in positions between them are called the *disparities*. The depth information of the point can then be recovered (*reconstruction*) by using the system setup parameters and the disparities. In this thesis, to facilitate the determination of the disparities, rectification of the geometry of virtual cameras is developed and implemented.

A geometry-based approach has been proposed to solve stereo vision rectification issue of the stereovision in this work which involves virtual cameras. The projection transformation matrices of a group of virtual cameras are computed by a unique geometrical ray sketching approach, with which the extrinsic parameters can be obtained accurately. This approach eliminates the usual complicated calibration process. Comparing the results of the geometry-based approach to the results of camera calibration technique, the former approach produces better results. This approach has also been generalized to a single-lens based multi-ocular stereovision system.

Next, an algorithm of segment-based stereo matching using cooperative optimization to extract the disparities information from stereo image pairs is proposed. This method combines the local method and the global method, which utilizes the favourable characters of the two methods such their computational efficiency and accuracy. In addition, the algorithm for multi-view stereo matching has been developed, which is generalized from the two views

stereo matching approach. The experimental results demonstrate that our approach is effective in this endeavour.

Finally, a triangulation algorithm was employed to recover the 3D depth of a scene. Note that the 3D depth can also be recovered from disparities as mentioned above. Therefore, this algorithm based on triangulation can also be used to verify the overall correctness of the stereo vision rectification and stereo matching algorithm.

To summarize, the main contribution of this thesis is the development of a novel stereo vision technique. The presented single lens prism based multi-ocular stereovision system may widen the applications of stereovision system; such as close-range 3D information recovery, indoor robot navigation / object detection, endoscopic 3-D scene reconstruction, etc.

LIST OF TABLES

Table 2.1 Block matching methods	23
Table 2.2 Summary of 3-D reconstruction three cases [10]	31
Table 3.1 The parameters of single-lens stereovision using biprism.....	46
Table 3.2 The values of parameters for bi-prism used in the experiment	58
Table 3.3 The descriptions of the columns in Table 3.4	64
Table 3.4 Results of conventional calibration method and geometrical method for obtaining stereo correspondence.....	65
Table 4.1 The parameters of tri-prism used in our setup.....	73
Table 4.2 The descriptions of the columns in Table 4.3	93
Table 4.3 The result of comparing calibration method and geometry method for obtaining stereo correspondence.....	94
Table 5.1 Percentages of bad matching pixels of reference images by five methods.....	113
Table 6.1 Percentages of bad matching pixels of disparity map obtained by the two methods compare with ground-truth	128
Table 6.2 Middlebury stereo evaluations on different algorithms, ordered according to their overall performance	136
Table 7.1 The results of two-view and multi-view stereo matching algorithm	155
Table 7.2 Recovered depth using binocular stereovision	161

LIST OF FIGURES

Figure 1.1 A perfectly undistorted, aligned stereo rig and known correspondence	3
Figure 1.2 Depth varies inversely to disparity	4
Figure 1.3 Description of the overall stereo vision technique of our thesis	6
Figure 2.1 Conventional stereovision system using two cameras.....	10
Figure 2.2 Modeling of two camera canonical stereovision system.....	11
Figure 2.3 A single-lens stereovision system using a glass plate.....	12
Figure 2.4 A single-lens stereovision system using three mirrors.....	12
Figure 2.5 Symmetric points from symmetric cameras	13
Figure 2.6 A single-lens stereovision system using two mirrors	13
Figure 2.7 The epipolar geometry	16
Figure 2.8 The geometry of converging stereo with the epipolar line (solid) and the collinear scan-lines (dashed) after rectification.....	18
Figure 2.9 (a) disparity-space image using left-right axes and; (b) another using left-disparity axes.....	26
Figure 3.1 Single-lens based stereovision system using bi-prism.....	33
Figure 3.2 Single-lens stereovision using optical devices.....	34
Figure 3.3 Pinhole camera model.....	35
Figure 3.4 Epipolar geometry of two views	37
Figure 3.5 Rectified cameras. Image planes are coplanar and parallel to baseline	38
Figure 3.6 Geometry of single-lens bi-prism based stereovision system (3D).....	44
Figure 3.7 Geometry of left virtual camera using bi-prism (top view).....	45
Figure 3.8 The relationship of direction vector of AB and normal vector of plane Π_1	49
Figure 3.9 The relationship of direction vector of AB and normal vector of plane Π_3	51
Figure 3.10 Rectification of virtual image planes	56

Figure 3.11 $\alpha = 6.4^0$, “robot” image pair (a) and rectified image pair (b)	60
Figure 3.12 $\alpha = 20^0$, “soap bottle” image pair (a) and rectified pair (b)	61
Figure 3.13 $\alpha = 45^0$ “cif” image pair (a) and rectified pair (b)	62
Figure 3.14 $\alpha = 10^0$, “Pet” image pair (a) and rectified pair (b)	63
Figure 4.1 Single-lens based stereovision system using tri-prism	67
Figure 4.2 Single-lens stereovision system using 3F filter	68
Figure 4.3 The structure of tri-prism	70
Figure 4.4 Geometry of left virtual camera using tri-prism	71
Figure 4.5 The workflow of determining the extrinsic parameters of virtual camera via geometrical analysis	72
Figure 4.6 Relationship of direction vector line PM	76
Figure 4.7 Illustration of direction vector of line MN	78
Figure 4.8 The virtual image plane π rotated to image plane π_1 about x -axis	80
Figure 4.9 The relationship of z' -axis and z -axis	81
Figure 4.10 The image plane π_1 rotates to image plane π_2 about y' -axis	82
Figure 4.11 Geometry of single-lens based on stereovision system using 4-face prism	86
Figure 4.12 Geometry of the single-lens stereovision system using 5-face prism	89
Figure 4.13 The image captured from trinocular stereovision and rectified images (robot) ...	91
Figure 4.14 The image captured from trinocular stereovision and rectified images	92
Figure 4.15 The images capture from four-ocular stereovision (“da” images)	95
Figure 4.16 The images capture from four-ocular stereovision and rectified images (“da” images)	96
Figure 5.1 The flow chart of obtaining depth map from stereo matching algorithm	99
Figure 5.2 Segmented by mean-shift method	103
Figure 5.3 Segmented by mean-shift method (using standard image)	103
Figure 5.4 Block diagram of the algorithm’s structure	110
Figure 5.5 Initial disparity maps by five methods (SAD, SSD, NCC, SHD, our method) ...	111

Figure 6.1 The flow chart of the estimated disparity plane parameters	121
Figure 6.2 Two type properties of plane	124
Figure 6.3 The flow chart for the procedure of merging the neighboring disparity plane	126
Figure 6.4 The results of disparity map obtained in each stage	127
Figure 6.5 Segments after implementation of mean-shift method	129
Figure 6.6 Final results of the disparity maps obtained by our algorithm (cooperative optimization)	133
Figure 6.7 “Robot” images: (a) Rectified image pair, (b) Robot image, which are extracted from rectified image in square, and (c) disparity map.....	134
Figure 6.8 “Pet” images: (a) rectified image pair (b) Pet image, which are extracted from rectified image in square, and (c) disparity map.....	135
Figure 6.9 “Fan” image: (a) “Fan” image and (b) disparity map	135
Figure 7.1 Collinear multiple stereo.....	139
Figure 7.2 The multi-view stereo pairs	143
Figure 7.3 Stereo images system.....	150
Figure 7.4 Triangulation with nonintersecting.....	150
Figure 7.5 Rectified cameras image planes	152
Figure 7.6 Tsukuba images: (a), (b), and (c) are Tsukuba images, (d) ground-truth map, (e) multi-view stereo matching algorithm result (local method), (f) multi-view stereo matching algorithm result (global method).....	154
Figure 7.7 The rectified “da” images.....	156
Figure 7.8 “da” images disparity map.....	156
Figure 7.9 “Pet” image depth recovery: (a) original image of pet, (b) the disparity map, and (c) depth reconstruction	157
Figure 7.10 “Fan” image depth recovery: (a) original image of pan, (b) the disparity map, and (c) depth recovery.....	158
Figure 7.11 “Robot” image depth recovery: (a) original image of robot, (b) the disparity map, and (c) depth recovery.....	159
Figure 7.12 “da” image depth recovery: (a) the disparity map of “da”, and (b) depth recovery	160

Figure 7.13 Several test points are selected in robot image	161
---	-----

LIST OF ABBREVIATIONS

3D/3-D	Three-dimension
2D/2-D	Two-dimension
CGI	Computer Generated Imagery
CCD	Charge-Coupled Devices
PPM	Perspective Projection Matrix
CCS	Camera Coordinate System
WCS	World Coordinate System
SVD	Singular Value Decomposition
HVS	Human Visual System
AD	Absolute intensity Differences
DSI	Disparity Space Image
SAD	Sum of Absolute Differences
ZSAD	Zero-mean Sum of Absolute Differences
LSAD	Locally scaled Sum of Absolute Differences
SSD	Sum of Squared Differences
SSSD	Sum of sums of absolute differences
ZSSD	Zero-mean Sum of Squared Differences
LSSD	Locally scaled Sum of Squared Differences
NCC	Normalized Cross Correlation
ZNCC	Zero-mean Normalized Cross Correlation
SHD	Sum of Hamming Distances
WTA	Winner-take-all
DP	Dynamic Programming
GC	Graph Cuts

CA	Cooperative Algorithms
NN	Neural Network algorithm
BP	Belief Propagation
BPASW	Biologically and Psychophysically inspired Adaptive Support Weights

LIST OF SYMBOLS

Baseline, i.e. the distance between the two camera optical centres:	λ
The disparity of the corresponding points between the left and right image:	d
The center of left image plane:	$c_l(o_{xl}, o_{yl})$
The center of right image plane:	$c_r(o_{xr}, o_{yr})$
The depth of object in world coordinate system:	Z
Effective real camera focal length:	f
Rotation matrix:	R
Translation vector:	T
The object point in world coordinate frame:	P
The point on the left image plane:	p_l
The point on the right image plane:	p_r
The optical center of camera:	C
World coordinate system:	(X_w, Y_w, Z_w)
Camera coordinate system:	(x_c, y_c, z_c)
Perspective projection matrix:	P_{ppm}
The intrinsic parameters:	M_{int}
The extrinsic parameters:	M_{ext}
The fundamental matrix:	F
The epipole of left image:	e_l
The epipole of right image:	e_r
The corner angle of the bi-prism:	α
The refractive index of the prism glass material:	n
The focal length of the virtual cameras:	f_v

Chapter 1 Introduction

1.1 Background

In computer vision, stereovision is a popular research topic due to new demands in various applications, notably, in security and defense. Stereovision is the extraction of 3D information from two or multiple digital images of a same scene captured by more than one CCD camera. Human beings have the ability to perceive depth easily through the stereoscopic fusion of a pair of images registered from the eyes. Therefore, we are able to perceive the three-dimensional structure/information of objects in a scene. Although the human visual system is still not fully understood, stereovision technique which models the way humans perceive range information has been developed to enable and enhance the extraction of 3D depth information. Stereovision is now widely used in areas such as automatic inspection, medical imaging, automotive safety, surveillance, and other applications. References [1-7] give a list of existing applications.

Over the years, the foundation of 3D vision has been developed continuously. According to Marr [8], the formation of 3D vision is as follows: *‘Form an image (or a series of images) of a scene, derive an accurate three-dimensional geometric description of the scene and quantitatively determine the properties of the objects in the scene’*. In other words, 3D vision formation consists of three steps: *Data Capturing, Reconstruction and Interpretation*. Barnard and Fischler [9] have proposed a different list of steps for the formation of 3D stereovision which include *camera calibration, stereo correspondence, and reconstruction*. For each of these steps, many methods have been developed. However, the search for effective and simple methods for each of the steps is still an active research area.

This thesis aims to study the reconstruction of a 3-dimensional scene, or also known as depth recovery, using a single-lens stereovision system using prism [21]. The present work reported in this thesis includes the development of the stereo rectification, stereo correspondence and 3-D scene reconstruction algorithms. This introductory chapter is divided into five sections. Section 1.1 provides the background of stereovision. Section 1.2 presents the problem descriptions, while the next section, Section 1.3 presents our motivation. Section 1.4 describes the scope of study and objectives of this research. The final section, Section 1.5, gives the outline of the entire thesis.

1.2 Problem descriptions

Stereo vision refers to the ability to infer information on the 3-D structure and distance of a scene from two or more images [10]. From a computational standpoint, a stereovision system must solve two problems. The first one is known as stereo correspondence, which consists of determining the corresponding points of the image points in one image (the left image, say) in the other image (right image in this case). The purpose of this process is really to determine the disparity between the two corresponding points which will be discussed in detail below. In addition, due to the occlusion problem, some parts of the scene are not visible in one of the images. Therefore, a stereovision system must also be able to determine which parts of the image at which the search of the corresponding points are not possible.

The second aspect of a stereovision system is to recover the depth of a scene/object, which is called reconstruction, or depth recovery. Our vivid perception of the 3-D world is due to the interpretation in the brain which gives the computed difference in retinal position, named as *disparity*, between the corresponding features of objects in a scene. The disparities of all the image points form the so-called disparity map which can be displayed as an image. If the

geometry of the stereovision system is known, the disparity map can be converted into a 3-D map (reconstruction). [10]

The two aforesaid problems of stereovision, stereo correspondence and reconstruction have been studied by many researchers [35, 63-74]. Figure 1.1 shows a parallel stereovision system. c_l and c_r are the centre points of the left and right image planes, O_l and O_r are the optical centers of left and right cameras, x_l and x_r are the coordinates of image points in left and right image plane, f is the focal length and λ is the baseline of the two cameras.

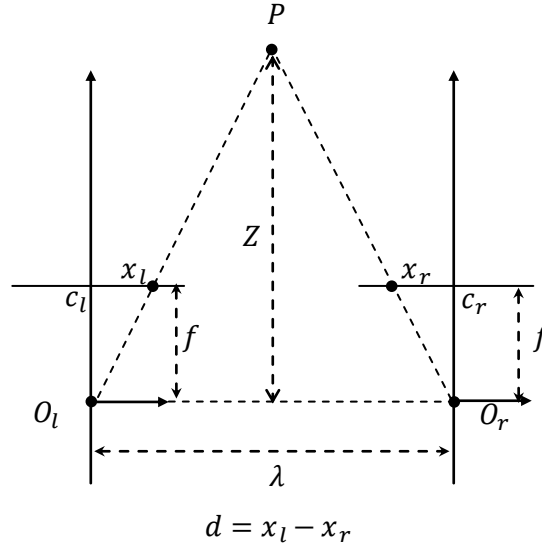


Figure 1.1 A perfectly undistorted, aligned stereo rig and known correspondence

The depth, Z , can be recovered from the geometry of the system as follows:

$$\frac{\lambda - (x_l - x_r)}{Z - f} = \frac{\lambda}{Z} \quad (1.1)$$

$$Z = \frac{f\lambda}{x_l - x_r} \quad (1.2)$$

Eq. (1.2) expresses the relationship of the depth with $x^l - x^r$.

Here, we let

$$d = x_l - x_r \quad (1.3)$$

where d denotes the disparity between the corresponding points between the left and right image.

We can also conclude from Eq. (1.2) that the depth is inversely proportional to the disparity. Thus, there is a nonlinear relationship between these two terms (see Figure 1.2).

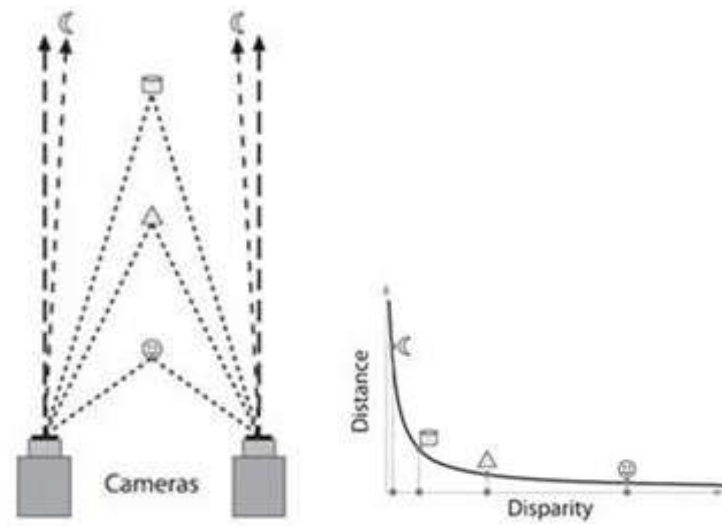


Figure 1.2 Depth varies inversely to disparity

To sum up, the stereovision work reported in this thesis will consist of the following areas:

- (1) Stereo rectification (Chapter 3 and 4)
- (2) Stereo correspondence (Chapter 5, 6 and 7)
- (3) Depth recovery (Chapter 7)

However, we have made the assumption that the captured images are free of distortion. We will follow these three steps in solving the stereo problems – depth recovery. The next section will present the motivation of our work reported in this thesis.

1.3 Motivation

The projection of light rays onto the retina of our eyes will produce a pair of images which are inherently two dimensional. However, based on this image pair, we are able to interact with the 3-D surrounding in which we are in. This ability implies that one of the functions of the human visual system is to reconstruct the 3-D structure of the world from a 2-D image pair. We shall develop algorithms to re-produce this ability using stereovision system. In our works, the said desired motivation consists of the three important aspects, stereo rectification, stereo correspondence, and depth recovery.

The complexity of the correspondence problem depends on the complexity of the scene. There are constraints (epipolar constraint [10], order constraint) and schemes that can help in reducing the number of false matches but there are still many unsolved problems in stereo correspondence. Some of these problems are:

- (1) Occlusion which may result in the failure on the searching of corresponding points.
- (2) Regularity and repetitive patterns in the scene may cause ambiguity in correspondence.

Finally, note that the accuracy of the 3D depth recovery or reconstruction depends heavily on the results of the stereo vision rectification and stereo correspondence.

1.4 Scope of study and objectives

The basis for stereovision is a single three-dimensional physical scene which is projected to a unique pair of images in two or multiple cameras. The first step of stereovision technique is *image acquisition* which usually employs two or more cameras to capture different views of a scene. When a point in the scene is projected into different locations on each image plane, there will be a difference in the position of its projections, which is called *disparity*. The depth recovery or 3D reconstruction of the point can be done by using the properties of the individual cameras, the geometric relationships between the cameras and the disparity. Figure 1.3 shows the overall stereovision setup and steps in this thesis. The works reported in this thesis, consisting of the steps shown in Figure 1.3, will follow closely the flow chart shown.

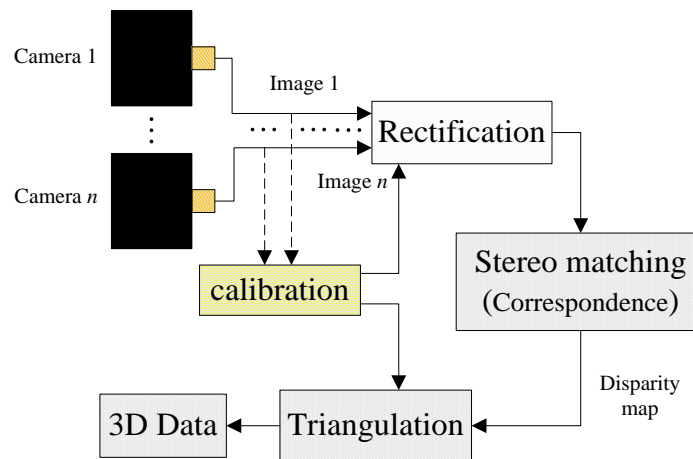


Figure 1.3 Description of the overall stereo vision technique of this thesis

The main objective of this work is to develop efficient methods in solving stereovision problem. More specifically, algorithms and strategies will be designed and implemented to recover 3-D depth of a given scene using a stereovision setup. The followings steps, each of which pertains to a specific problem, will be dealt with. The cohesive whole formed by the solutions of the problems presented in the steps would represent the objective of this thesis.

- (1) Investigate the basis of a single-lens prism based stereovision system developed by Lim and Xiao [21]. Knowledge gained here would be the use of this novel system and its use in calibrating the system to determine the intrinsic and extrinsic parameters.
- (2) Explore a geometry-based method to rectify the image pairs captured by the single-lens based stereovision system.
- (3) Develop a stereo correspondence algorithm for the image pairs, by combining local and global methods to solve the correspondence problem. In addition, this algorithm is extended to solve the multi-view stereo correspondence problem.

The results obtained from this study form a theoretical foundation for the development of a compact 3D stereovision system. Moreover, this research may contribute to a better understanding of the mechanism of the stereovision system as the nature of our method is to analyze the light ray sketching of the cameras. The next section will present the outline of this thesis.

1.5 Outline of the thesis

In this thesis, the algorithms involved in stereovision are studied and developed to recover the depth of a scene in 3-dimensions. The outline of the entire thesis is as follows:

Chapter 2 presents the literature review about stereovision which includes stereovision systems, camera calibration, epipolar geometry constraints, rectification algorithm, stereo correspondence algorithms and depth reconstruction.

Chapter 3 describes and discusses stereo vision rectification based on single-lens binocular stereo vision. A geometry-based approach is proposed to determine the extrinsic parameters of the virtual cameras with respect to the real camera. The parallelogram and refraction rules

are applied to determine the geometrical ray; this is followed by the computation of the rectification transformation matrix which is applied on the captured images using the single-lens stereovision system.

In Chapter 4, stereovision rectification based on trinocular and mutli-ocular is introduced. The geometry-based approach is extended to solve the multi-view stereo rectification problem.

Chapter 5 discusses part of the proposed stereo correspondence algorithm using the local method. In this chapter, image segmentation and initial disparity map acquisition are presented.

Chapter 6 presents the second part of the stereo matching algorithm using the global method. In this chapter, the steps of disparity plane estimation and cooperative optimization of energy function are introduced.

In Chapter 7, the algorithms for multi-view stereo matching and 3D depth recovery are proposed. The algorithm of stereo matching is applied to multi-view to solve correspondence problem.

Finally, the conclusions and future works are presented in Chapter 8.

Chapter 2 Literature review

In this chapter, recent works pertaining to stereovision techniques are reviewed. They include the algorithms of rectification, calibration, stereo correspondence and depth recovery. This chapter is divided into seven sections. Section 2.1 reviews various stereovision systems developed earlier by researchers. Section 2.2 presents camera calibration technique while the next section describes the epipolar geometry constraints, which are important in stereo correspondence. Section 2.4 gives a review on the existing rectification algorithms and Section 2.5 presents the stereo matching algorithms to solve stereo correspondence problems. Section 2.6 discusses various 3-D reconstruction techniques. The final section, Section 2.7 summarizes the reviews done in this chapter.

2.1 Stereovision systems

Research on the recovery and recognition of 3-D shapes or objects in a scene has been undertaken by using a monocular image and with multiple views. Depth perception by stereo disparity has been studied extensively in stereovision. The stereo disparity between the two images captured from two distinct viewpoints is a powerful cue to 3-D shapes and pose estimation. To recover a 3-D scene from a pair of stereo images of the scene, correspondences problem must first be resolved [10]. We shall present several configurations of stereovision systems below, and the various pertinent parameters are also defined and explained.

Conventionally, stereovision system requires two or more cameras to capture images of a scene from different orientations to obtain the disparity for the purpose of depth recovery. Figure 2.1 shows the conventional stereovision system using two cameras.

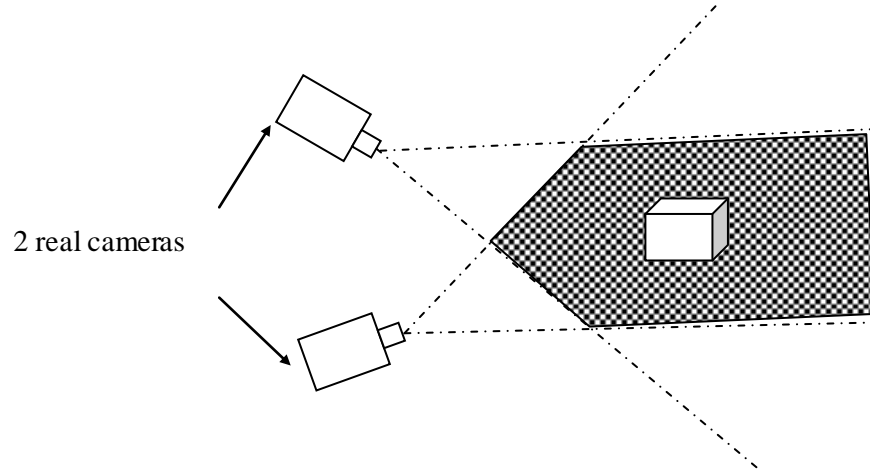


Figure 2.1 Conventional stereovision system using two cameras

Another simple canonical stereovision system employing two parallel cameras is shown in Figure 2.2. In this setup, the focal lengths of the two cameras are assumed to be the same. Furthermore, the two optical centres are assumed to be in the same X - Z plane. The coordinates of the scene point could be obtained from figure 2.2 and are shown below:

$$X_w = \frac{\lambda(x_l + x_r)}{x_l - x_r} ; Y_w = \frac{\lambda(y_l + y_r)}{x_l - x_r} ; Z_w = \frac{\lambda f}{x_l - x_r} \quad (2.1)$$

where λ is the length of the baseline connecting the two optical centers and f the focal length of both the cameras which are assumed to be the same.. The remaining symbols are defined in Figure 2.2. The disparity is defined as $(x_l - x_r)$, which is very important in depth recovery as can be seen from the expression for Z_w in equation (2.1). In addition, the two points on the two different image planes (p_l and p_r) must be from the same point in the scene, and they are known as corresponding points. A main bulk of work in 3-D depth recovery is in the search of corresponding points from the two captured images. This is in fact known as Correspondence Search Problem in stereo vision. In this simple and ideal system, it is obvious to note that the

corresponding points lie on the same scan lines in the two images, and are parallel to the baseline of the system. Thus this configuration simplifies the correspondence search problem.

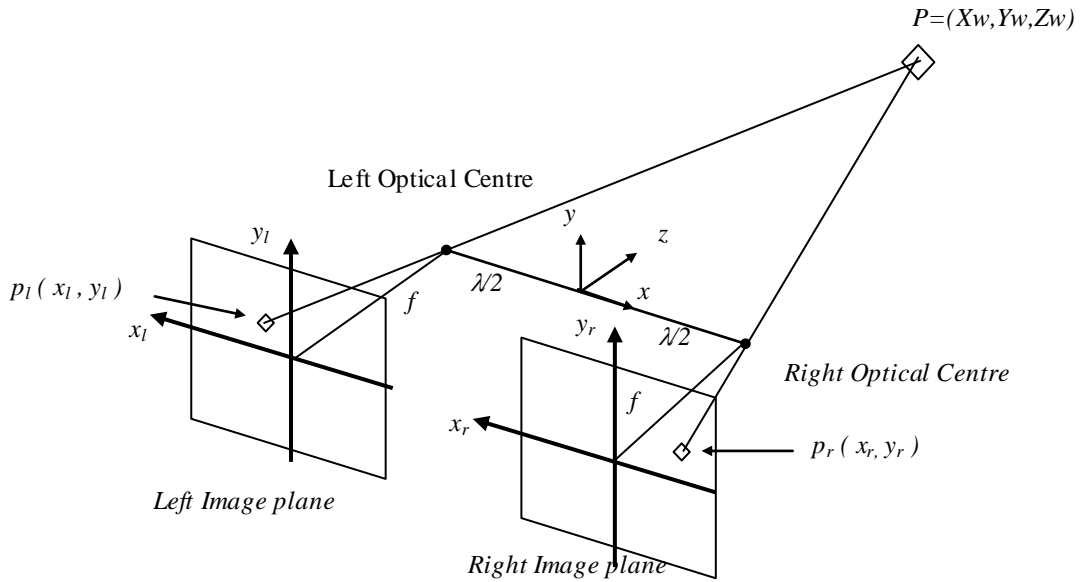


Figure 2.2 Modeling of two camera canonical stereovision system

The conventional stereovision systems have the advantages of simpler setup and ease in implementation. However, the difficulty in synchronized capturing of the image pairs by the two cameras and the cost of system make them less attractive. Therefore, single-lens stereovision systems [15] are explored by researchers to solve these short-comings.

In the past few decades, there were various single-lens stereovision systems proposed to potentially replace the conventional two cameras system with some significant advantages such as lower hardware cost, compactness, and reduction in computational load.

Single-lens stereovision system with optical devices was first proposed by Nishimoto and Shirai [16]. They use a glass plate which is positioned in front of a camera and the glass plate is free to rotate. The rotation of the glass plate to different angular positions allows a pair of stereo images to be captured (see Figure 2.3). The main disadvantage of this system is that the

disparities between the image pairs are small. Teoh and Zhang [17] further improved the idea of the single-lens stereovision camera with the aid of three mirrors. Two of the mirrors are fixed at 45 degrees at the top and bottom, and the third mirror can be rotated freely in the middle between the two said mirrors (see Figure 2.4). Two shots can be taken with the third mirror placed in positions parallel to the two fixed mirrors in separate instances. Francois et al. [18] further refined the concepts of stereovision from a single perspective to a mirror symmetric scene and concluded that a mirror symmetric scene is equivalent to observing the scene with two cameras, and all the traditional analysis tools of binocular stereovision can then be applied (Figure 2.5). The main problem of mirror based single-lens stereovision systems shown above is that they can only be applied to static scenes as the stereo image pairs are captured by two separate shots. This problem was overcome by Gosthasby and Gruver [19] whose system captured image pairs by the reflections from the two mirrors (Figure 2.6).

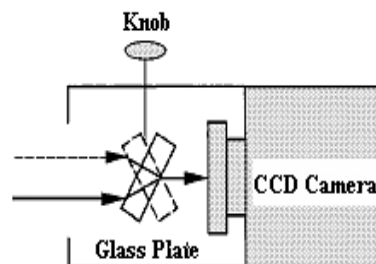


Figure 2.3 A single-lens stereovision system using a glass plate

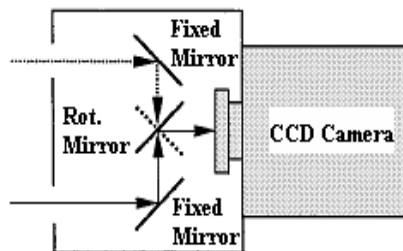


Figure 2.4 A single-lens stereovision system using three mirrors

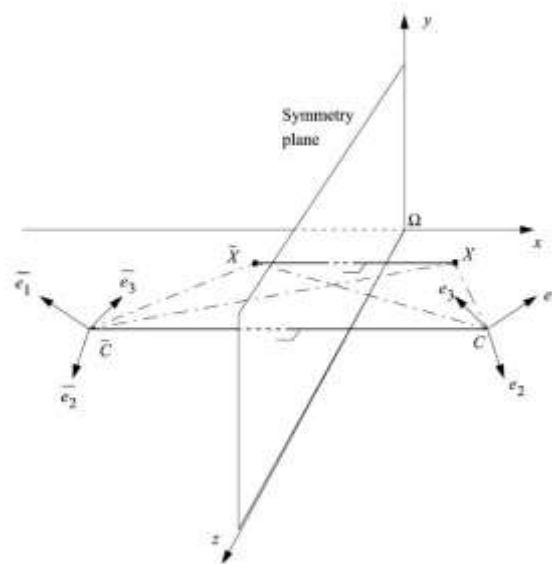


Figure 2.5 Symmetric points from symmetric cameras

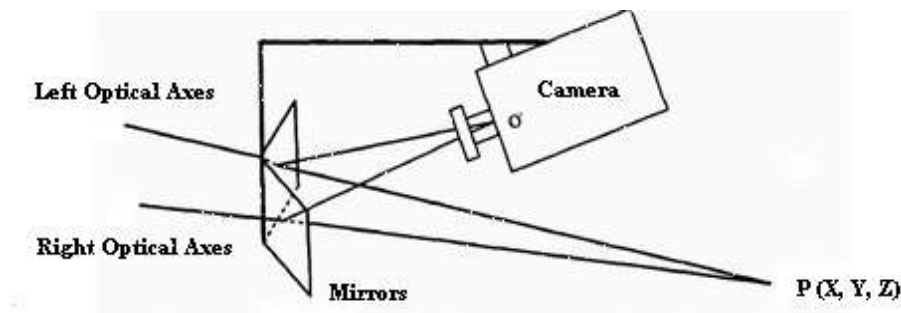


Figure 2.6 A single-lens stereovision system using two mirrors

Lee and Kweon [20] proposed a single-lens stereovision system using a bi-prism which was placed in front of a camera. Stereo image pairs were captured on the left and right halves on the image plane of the camera due to refraction of light rays through the prism. However, no detailed analysis was provided by them. Later, Lim and Xiao [21, 22] proposed a similar system and extended the study to include the use of multi-face prism. They also proposed the idea of calibrating the virtual cameras. One significant advantage of this prism based virtual stereovision system relative to the conventional two or multiple camera stereovision system is that only one camera is required, hence, fewer camera parameters need to be handled. In addition, the camera-synchronization problem in image capturing is eliminated automatically.

This one-camera simple setup can easily be modeled by a direct geometrical analysis of ray sketching.

The single-lens prism based stereovision system also has many other advantages [21], including:

- 1) The new setup will significantly reduce the cost of building a multi-camera stereovision system;
- 2) The compact setup will minimize the space required;
- 3) It has lesser system parameters and it is easy to implement, especially for the approach of determining the system parameters using geometrical analysis of ray sketching; and
- 4) The system eliminates the necessity in synchronization when capturing more than one image.

In fact, our works developed in this thesis are based on this simple single-lens prism based stereovision system.

2.2 Camera calibration

After setting up the stereovision system, the next task is to calibrate the various components of the system, such as the camera, fixtures, optical devices, etc. and their physical locations. Camera calibration is an important process to determine the intrinsic and extrinsic parameters of the system. The intrinsic parameters are inherent in a camera system, which normally include the effective focal length, lens distortion coefficients, scaling factors, position and orientation of the coordinates of the camera. The extrinsic parameters include the translation

and orientation information of the camera or image frame with respect to a specified world coordinate system.

The accuracy of the results of camera calibration will directly affect the performance of a stereovision system. Therefore, great efforts are spent to deal with this challenge. Based on the techniques used, camera calibration methods can be classified into 3 categories: linear transformation method, direct non-linear minimization method, and Hybrid method.

(1) Linear transformation methods. In these methods, the objective equations are linearized from the relationship between the intrinsic and extrinsic parameters [23, 24]. Therefore, the parameters are only the solutions of linear equations.

(2) Direct non-linear minimization methods. These methods use the interactive algorithms to minimize the residual errors of a set of equations which can be established directly from the relationship between the intrinsic and extrinsic parameters. They are only used in the classical calibration techniques [25, 26].

(3) Hybrid methods. These methods make use of the advantages of the two previous categories. Generally, they comprise two steps: the first step involves the linear equations to solve for most of the camera parameters; the second step employs a simple non linear optimization to obtain the remaining parameters. These calibration techniques could be used on different camera models with different lens-distortion models. Therefore, they are widely studied and used in recent works [27, 28, 29].

2.3 Epipolar geometry constraints

A concept in stereovision, known as epipolar geometry [10], is illustrated in Figure 2.3. The figure shows two pinhole cameras, with their optical centers, located at O_l and O_r . The image

planes, are shown as π_l and π_r . The focal lengths are denoted by f_l and f_r . Each camera is defined with a 3-D reference frame, the origin of which coincides with the optical center, and the Z-axis is along the optical axis. The vectors $\mathbf{P}_l = [X_l, Y_l, Z_l]^T$ and $\mathbf{P}_r = [X_r, Y_r, Z_r]^T$ refer to the same 3-D point, P , thought of as a vector with respect to the left and right world coordinate frame respectively. The vectors $\mathbf{p}_l = [x_l, y_l, z_l]^T$ and $\mathbf{p}_r = [x_r, y_r, z_r]^T$ refer to the projections of P onto the left and right image planes, respectively, and are expressed in the corresponding reference frame (Figure 2.7). Thus, for all the image points, $z_l = f_l$ or $z_r = f_r$.

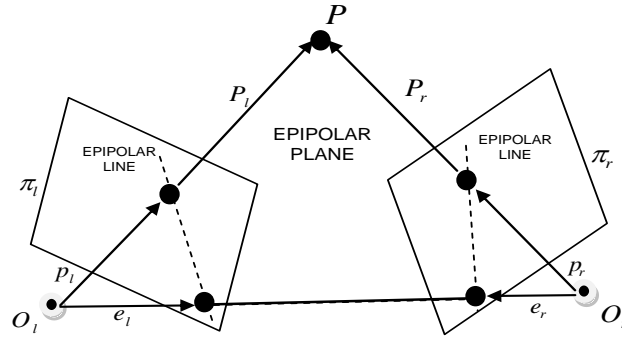


Figure 2.7 The epipolar geometry

The reference frames of the left and right cameras are related by the extrinsic parameters. Their relationship can be defined by a rigid transformation in 3-D space by a translation vector, $\mathbf{T} = (\mathbf{O}_r - \mathbf{O}_l)$, and a rotation matrix, \mathbf{R} . Given a point P in space, the relation between \mathbf{P}_l and \mathbf{P}_r can be written as $\mathbf{P}_r = \mathbf{R}(\mathbf{P}_l - \mathbf{T})$.

The name epipolar geometry is used because the points at which the line goes through the centers of projection intersects the image planes (Figure 2.7) are called epipoles. We denote the left and right epipoles by e_l and e_r respectively.

The relation between a point in 3-D space and its projections is described by the usual equations of perspective projection, in vector form:

$$\mathbf{p}_l = \frac{f_l}{Z_l} \mathbf{P}_l \quad (2.2)$$

and

$$\mathbf{p}_r = \frac{f_r}{Z_r} \mathbf{P}_r \quad (2.3)$$

Epipolar geometry defines a plane (epipolar plane) which is formed by P , O_l , and O_r . This plane intersects each image at a line, called epipolar line (see Figure 2.7). Considering the triplets, \mathbf{p}_l , \mathbf{p}_r and P , given \mathbf{p}_l , P can be any point on the ray from O_l through \mathbf{p}_l . Since the dash line in the right image (see Figure 2.7) is the epipolar line through the corresponding point, \mathbf{p}_r , the correct match must lie on the epipolar line. This important fact is known as the epipolar constraint. It establishes a mapping between points in the left image and lines in the right image and vice versa.

Thus, once the epipolar constraint is established, we can restrict the search for the match of \mathbf{p}_l , along the corresponding epipolar line. The search for correspondences is thus reduced to a 1- D problem. Alternatively, the same knowledge can be used to verify whether or not a candidate match lies on the corresponding epipolar line. This is usually the most effective procedure to reject false matches due to occlusions.

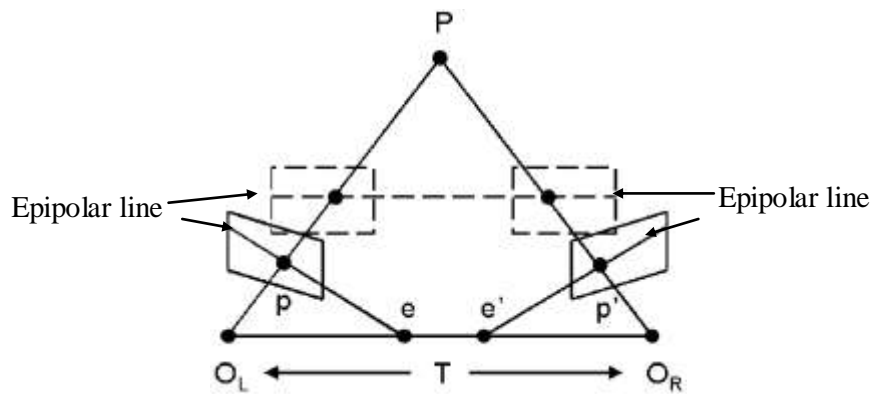


Figure 2.8 The geometry of converging stereo with the epipolar line (solid) and the collinear scan-lines (dashed) after rectification

The conventional converging stereovision system is shown in Figure 2.8. There are two epipolar lines, one in each of the two image planes (ep and $e'p'$). In this configuration, an epipolar line is not along a horizontal scan-line, but inclined at an angle to it. The search of a corresponding point in the left image (say), is along the epipolar line $e'p'$ at the right image, and vice-versa. Searching the corresponding point on an inclined line could be labourious, and it would be easier to conduct the search along a horizontal scan line. We shall use a rectification technique, reported in [10, 30] such that the epipolar lines are made to be along a horizontal scan lines of the images. This will facilitate the correspondence search process and will reduce both the computational complexity and the likelihood of false matches. In this thesis, we will be exploring the rectification technique for this reason.

2.4 Review of rectification algorithms

The objective of rectification has been mentioned in the previous section. It can essentially be viewed as a process to transform the image points on two non-coplanar image planes to be on two coplanar image planes. This will ensure that the two epipolar lines become collinear and are along a horizontal scan line across the two images. The correspondence search will be greatly simplified as reported in [34].

In the past, the rectification process in stereovision was primarily achieved using optical techniques [36]; recently the techniques have been replaced by software means. In essence, a single linear transformation to each image planes is designed and implemented using software, the transform effectively rotates both cameras until their image planes are coplanar ([35, 37, 12, 38]). Such techniques are often referred to as planar rectification. The advantages of this linear approach include mathematically simple, fast and able to preserve image features such as straight lines. However, these techniques might not be easily applied in more complex situations.

Rectification is a classical issue of stereo vision. However, limited numbers of methods exist in the computer vision literature. It can generally be classified into uncalibrated rectification and calibrated rectification. The first work on uncalibrated rectification called “matched-epipolar projection” is presented by Gupta [12], and followed by Hartley [37], who tidied up the theory. He uses the condition that one of the two collinear should be close to a rigid transformation in the neighborhood of a selected point, while the remaining degrees of freedom are fixed by minimizing the distance between corresponding points (disparity). Al-Shalfan et al. [39] presented a direct algorithm to rectify pairs of uncalibrated images: while Loop and Zhang proposed a technique to compute rectification homographies for stereo vision [13]. Isgro` and Trucco presented a robust algorithm performing uncalibrated rectification which does not require explicit computation of the epipolar geometry [40]. Later, Hartley [37, 42] gave a mathematical basis and a practical algorithm for the rectification of stereo images from different viewpoints [37, 43]. Some of these works also concentrate on the issue of minimizing the rectified image distortion. We do not address this problem in this thesis because distortion is less severe than in the weakly calibrated case.

For the calibrated rectification algorithm, Fusiello et al. presented a compact algorithm to rectify calibrated stereo images [44]. Ayache and Lustman [45] introduced a rectification

algorithm, in which a matrix satisfying a number of constraints is handcrafted. The distinction between necessary and arbitrary constraints is unclear in their case. Some authors reported rectification techniques they have developed under restrictive assumptions; for instance, Papadimitriou and Dennis [46] assumed a very restrictive geometry (parallel vertical axes of the camera reference frames). Ayache and Hansen [49] presented a technique for calibrating and rectifying image pairs or triplets. In their case, a camera matrix needs to be estimated, therefore the algorithm works for calibrated cameras. Shao and Fraser also developed a rectification method for calibrated trinocular cameras [50]. Point Grey Research Inc [51] used three calibrated cameras for stereo vision after rectification. These rectification algorithms for triplet images or trinocular images only work for calibrated stereovision systems.

In this thesis, we propose a geometry-based approach for rectification problem based on single-lens stereovision system using bi-prism and multi faced-prism. The advantages of single-lens stereovision system using prism have been introduced in Section 2.1. Compare with conventional method which requires the complicated calibration process, our proposed approach only requires several points on the real image to determine all the required system parameters of our virtual stereovision system. After the virtual cameras calibration, the rectification transformation matrix is determined to rectify the image planes of the virtual cameras.

2.5 Stereo correspondence algorithms

In practice, we are given two or more images; we have to compute the disparities from the information contained in these images. The correspondence problem consists of determining the locations in each camera image that are the projection of the same physical point in space. No general solution for correspondence problem exists, due to ambiguous matches due to occlusion, lack of texture, etc.) Assumptions, such as image brightness constancy and surface

smoothness are commonly made to render the problem tractable. In this section, we review several algorithms for stereo correspondence.

Daniel and Richard [14] described the detail of the taxonomy of stereo correspondence algorithm. It can be classified into local methods and globe methods. Local methods can be very efficient, but they are sensitive to ambiguous regions in images (e. g., occlusion regions or regions with uniform texture). Global methods can be less sensitive to these problems since global constraints provide additional support for regions which are difficult to be matched locally. However, these methods are more computationally expensive.

(1) Local Methods

In this section, we compare several local correspondence algorithms in terms of their performance and efficiency. These methods fall into three broad categories: gradient methods, and feature matching method and block matching method.

(a) Gradient Method

Gradient method or optical flow can be applied to determine small local disparities between two images by formulating a differential equation relating motion and image brightness. These methods are applicable under the assumption that as the time varies, the image brightness (intensity) of points does not change as they move in the image. In other words, the change in brightness is entirely due to motion [30, 54]. If the image intensity $E(x, y, t)$ of points (x, y) is a continuous and differentiable function of space and time, and if the brightness pattern is locally displaced by a distance (dx, dy) over a time period dt , then the gradient method can be mathematically expressed as:

$$E(x, y, t) = E(x + dx, y + dy, t + dt)$$

$$\frac{dE}{dt} = 0$$

$$\frac{dE}{dt} = E_x V_x + E_y V_y + E_t = \nabla E^T (V_x, V_y) + E_t = 0 \quad (2.5)$$

where E denotes the intensity, ∇E and E_t are the spatial image intensity derivative and the temporal image intensity derivative, respectively. Among them, ∇E and E_t are known parameters which can be measured from the images while (V_x, V_y) are the unknown optical flow components $(\frac{dx}{dt}, \frac{dy}{dt})$ in the x and y directions, respectively.

In summary, gradient-based methods can only work when the 2D motion is “small” so that the derivative can be computed reliably. Preferably, block matching and feature matching algorithm should be used to compute the 2D motion when the motion is “large”.

(b) Feature Matching Method

Given a stereo image pair, feature-based methods match features in the left image to those in the right image. Feature matching methods received significant attention as they are insensitive to depth discontinuities and insensitive to regions of uniform texture by limiting the regions of support to specific reliable features in the images. Venkateswar and Chellappa [55] discussed the hierarchical feature matching where the matching starts at the highest level of the hierarchy (surfaces) and proceeds to the lowest ones (lines) because higher level features are easier to match due to fewer numbers and more distinct in form. The segmentation matching introduced by Todorovic and Ahuja [56] aims to identify the largest part in one image and its match in another image having the maximum similarity measure defined in terms of geometric and photometric properties of regions (e.g., area, boundary, shape and color), as well as regions topology.

(c) Block Matching Method (Area-Based Method)

Block matching methods (area-based method) seek to find the corresponding points on the basis of correlation (similarity) between the corresponding areas in the left and right images [10]. It searches for maximum match score or minimum error over a small region. Moreover, the epipolar geometry is quite efficient for block matching because it can reduce the dimension of the corresponding point search. Table 2.1 shows the block matching methods.

Table 2.1 Block matching methods

MATCH	FORMULA METRIC
Sum of Absolute Differences (SAD)	$\sum_{(i,j) \in W} I_1(i,j) - I_2(x+i, y+j) $
Zero-mean Sum of Absolute Differences (ZSAD)	$\sum_{(i,j) \in W} I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i, y+j) + \bar{I}_2(x+i, y+j) $
Locally scaled Sum of Absolute Differences (LSAD)	$\sum_{(i,j) \in W} I_1(i,j) - \frac{I_1(i,j)}{\bar{I}_2(x+i, y+j)} I_2(x+i, y+j) $
Sum of Squared Differences (SSD)	$\sum_{(i,j) \in W} (I_1(i,j) - I_2(x+i, y+j))^2$
Zero-mean Sum of Squared Differences (ZSSD)	$\sum_{(i,j) \in W} (I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i, y+j) + \bar{I}_2(x+i, y+j))^2$
Locally scaled Sum of Squared Differences (LSSD)	$\sum_{(i,j) \in W} (I_1(i,j) - \frac{I_1(i,j)}{\bar{I}_2(x+i, y+j)} I_2(x+i, y+j))^2$

Normalized Cross Correlation (NCC)	$\frac{\sum_{(i,j) \in W} I_1(i,j) \cdot I_2(x+i, y+j)}{\sqrt{\sum_{(i,j) \in W} I_1^2(i,j) \cdot \sum_{(i,j) \in W} I_2^2(x+i, y+j)}}$
Zero-mean Normalized Cross Correlation (ZNCC)	$\frac{\sum_{(i,j) \in W} (I_1(i,j) - \bar{I}_1(i,j)) \cdot (I_2(x+i, y+j) - \bar{I}_2(x+i, y+j))}{\sqrt{\sum_{(i,j) \in W} (I_1(i,j) - \bar{I}_1(i,j))^2 \cdot \sum_{(i,j) \in W} (I_2(x+i, y+j) - \bar{I}_2(x+i, y+j))^2}}$
Sum of Hamming Distances (SHD)	$\sum_{(i,j) \in W} I_1(i,j) \text{ bitwise XOR } I_2(x+i, y+j)$

The sum of Absolute Differences (SAD) is one of the simplest similarity measures which is calculated by subtracting pixels within a square neighborhood between the reference image I_1 and the target image I_2 followed by the aggregation of absolute differences within the square window, and optimization with the winner-take-all (WTA) strategy [57]. If the left and right images match exactly, the resultant will be zero.

In Sum of Squared Differences (SSD), the differences are squared and aggregated within a square window and later optimized by WTA strategy. This measure has a higher computational cost compared to SAD algorithm as it involves numerous multiplication operations.

Normalized Cross Correlation is even more complex compared to both SAD and SSD algorithms as it involves numerous multiplication, division and square root operations.

Sum of Hamming Distances is normally employed for matching census-transformed images (can be used on images that have not been census transformed) by computing bitwise-XOR of the values in the left and right images within a square window. This step is usually followed by a bit-counting operation which results in the final Hamming distance score.

(2) Global Methods

As stated above, global correspondence methods exploit non-local constraints in order to reduce sensitivity to local regions in the image that fails to match, due to occlusion, uniform texture, etc. The use of these constraints makes the computational cost of global matching significantly greater compared to the local matching method.

Global algorithms first make explicit smoothness assumptions of the disparity map and then calculate the optimized matching globally by minimizing an energy function. Generally speaking, the global energy function contains two parts, smoothness energy part and data energy part, as follow:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \quad (2.6)$$

The data term, $E_{data}(d)$, measures how well the disparity function d agrees with the input image pair. The smoothness term $E_{smooth}(d)$ encodes the smoothness assumptions made by the algorithm. λ is a weight of the smoothness term.

We here review the global algorithms such as dynamic programming, graph cuts, cooperative algorithms, neural network algorithm and other global algorithms in detail.

(a) Dynamic Programming (DP)

Dynamic programming is a mathematical method that reduces the computational complexity of optimization problems by decomposing them into smaller and simpler sub-problems [58]. A global cost function is computed in stages, with the transition between stages defined by a set of constraints. For stereo matching, the epipolar monotonic ordering constraint allows the global cost function to be determined as the minimum cost path through a disparity space image (DSI). The cost of the optimal path is the sum of the costs of the partial paths obtained

recursively. The local cost functions for each point in the DSI may be defined using one of the area-based methods. There are two ways to construct a DSI, which is shown in Figure 2.9.

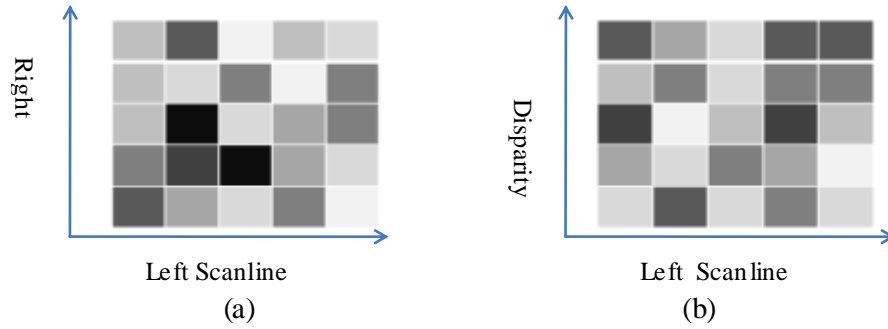


Figure 2.9 (a) disparity-space image using left-right axes and; (b) another using left-disparity axes

In Figure 2.9, the intensities shown represent the respective costs of potential matches along the scan-lines, with lighter intensities having lower cost. First, the axes may be defined as the left and right scanlines, as in the case Ohta and Kanade [59] and Cox et al. [60]. For this case, dynamic programming is used to determine the minimum cost path from the lower left corner to the upper right corner of the DSI. The second method for constructing a DSI is to define the axes as the left scanline and the disparity range (Figure 2.9(b)), as reported by Intille and Bobick [61]. In this case, dynamic programming is used to determine the minimum cost path from the first column to the last column (see Figure 2.9 (b)). Baker [62] proposed a dynamic programming method that first computes disparities independently for each scanline and then detects and makes corrections to those that violate the inter-scanline consistency constraints. Ohta and Kanade [59] proposed to integrate the inter-scanline constraints into the match process by minimizing the sum of costs over two-dimensional regions defined as intervals between vertical edges. The local cost function for each interval is defined as the variance of the pixel intensities in that interval. Belhumeur [63] proposed a two-stage approach, first computing intra-scanline solutions using dynamic programming and then smoothing disparities between scanlines. This smoothing is done by taking each three adjacent scanlines, fixing the disparities of the outer two, and then re-computing the optimum solution for the

middle scanline using dynamic programming. One of the principal advantages of dynamic programming is that it provides global support for local regions that lack texture which would be matched incorrectly otherwise. These local regions present little difficulty for a global search since any cost function in these regions is low. Another problem that global search seeks to resolve is occlusion. This is more difficult since a cost function applied near an occlusion boundary is typically high. Methods for dealing with this difficulty have been proposed in [63] and [64]. These methods replace matching costs at occlusion boundaries with a small fixed occlusion cost.

(b) Graph Cuts (GC)

Solving the problem with dynamic programming algorithm is not an effective integration of horizontal and vertical continuity constraints. Many approaches have been proposed to improve this situation while maintaining the dynamic programming framework. However, they do not fully exploit the two dimensional coherence constraints. An alternative approach that exploits these constraints is to cast the stereo matching problem as one of finding the maximum flow in a graph [58].

Many approaches have been developed for its efficient solution. Zhao [65], and Thomos et al. [66] use the well-known preflow-push lift-to-front algorithm. The complexity of this algorithm is $O(N^2 D^2 \log(ND))$, where N is the number of pixels in the image and D is the image resolution, which is significantly greater than that of dynamic programming algorithms. However, the average observed time reported by Roy and Cox [67] is $O(N^{1.2} D^{1.3})$, which is much closer to that of dynamic programming. One limitation of the left-to-front algorithm is that the classical implementations require significant memory resources, making this approach cumbersome for use with large images. Thomos et al. [66] have developed an

efficient data structure that reduces the memory requirements by a factor of approximately four, making this algorithm more manageable for large data sets.

Recent work on graph cuts has produced both new graph architectures and energy minimization algorithms. Boykov and Kolmogorov [68] have developed an approximate Ford-Fulkerson style augmenting paths algorithm, which is much faster in practice than the standard push-re-label approaches. Kolmogorov and Zabih [69] propose a graph architecture in which the vertices represent pixel correspondences (rather than pixels themselves) and impose uniqueness constraints to handle occlusions. These recent graph cut methods have been shown to be among the best performers in [14].

(c) Cooperative Algorithms (CA)

The cooperative optimization is a newly discovered general optimization method to address the hard optimization problems [70]. It has been found in the experiments reported in [71] that cooperative optimization has achieved remarkable performances at solving a number of real-world *NP*-hard problems with the number of variables ranging from thousands to hundreds of thousands. The problems span several areas, proving its generality and power. For example, cooperative optimization algorithms have been proposed for DNA image analysis [71], shape from shading [72] and stereo matching [73]. In the second case, it significantly outperformed the classic simulated annealing in finding the global optimal solutions. In the third case, its performance is comparable with graph cuts in terms of solution quality, and is twice as fast as graph cuts in software simulation using the common evaluation framework for stereo matching [14]. In the fourth case, it is ten times faster than graph cuts and has reduced the error rate by two to three times. In all these cases, its memory usage is efficient and fixed with simple, regular, and fully scalable operations. All these features make it suitable for parallel hardware implementations.

Cooperative algorithms, inspired by computational models of human stereo vision, were among the earliest methods proposed for disparity computation [74, 75]. Such algorithms iteratively perform local computations, but use nonlinear operations that result in an overall behavior similar to global optimization algorithms. In fact, for some of these algorithms, it is possible to explicitly state a global function that is being minimized [76]. Recently, a promising variant of Marr and Poggio's original cooperative algorithm has been developed [77]. Cooperative approaches [77] compute matching scores locally using match windows. Nevertheless, they show "global behavior" by refining the correlation scores iteratively using the uniqueness and continuity constraints. Zhang and Kambhamettu [78] take advantage of image segmentation in the calculation of the initial matching scores. Furthermore, they exploit the results of the segmentation in their choice of local support area, preventing the support area from overlapping a depth discontinuity.

(d) Neural Network algorithm (NN)

The principle of this method is that, according to the type of network, matching cost function and constraints are translated into minimization optimization using the iterative learning algorithm. Network dynamic process realizes the minimization of a number of constraints. The trained network can almost achieve real-time measurement to obtain three-dimensional information. At present, a number of this kind of algorithm uses Hopfield network. Ruichek realized stereo matching in obstacle detection by Hopfield network [79]. Hua [80] realized that by directly mapping color image based on the competition-cooperative network cannot express the whole 2D disparity map changing process. To meet the overall best match, constructing 3D Hopfield networks is a more meaningful job.

(e) Other global methods

While dynamic programming, and more recently graph cuts have been the most often exploited energy minimization methods for global stereo matching, a number of other approaches have been used as well. Two of the most notable ones are nonlinear diffusion and coordination algorithm [60]. Shah [81], Scharstein and Szeliski [82], and Mansouri[83] use various models for non-uniform diffusion, rather than using fixed-size, rectangular windows. Coordination algorithm comes from human visual computing model; its nonlinear iterative operation is similar to global algorithm in the whole image.

Another class of global methods seeks to reconstruct a scene without explicitly establishing correspondences. Fua and Leclerc [84] model the scene as a mesh that is iteratively updated to minimize an objective function. Faugeras and Kriven [85] proposed a similar method that models the scene using level sets. Kutulako and Seitz [86] presented the scene as a volume and proposed a space carving method to refine the surface. While these methods may be applied to binocular stereo vision, the object-centered representations are more powerful when exploiting constraints from multiple views of the scene to reduce sensitivity to view-dependent effects.

Recently, global methods are applied into image regions calculated by the Mean-Shift color segment algorithm [87, 88]. They are based on the assumption that the scene structure can be approximated by a set of non-overlapping planes in the disparity space and that each plane is coincident with at least one homogeneous color segment in the reference image. Generally speaking, the segment-based stereo matching has four steps [89]. Firstly, segment the reference image using robust segmentation method; secondly, obtain initial disparity map using local match method; thirdly, a plane fitting technique is employed to obtain the disparity planes; and finally, an optimal disparity plane (optimal labeling) is approximated

using Belief Propagation (BP) [89], graph cut optimization, or cooperative optimization method. We also use these steps in our thesis to solve stereo correspondence problem.

2.6 Stereo 3-D reconstruction

After solving the correspondence problem, the 3-D reconstruction can be obtained depending on the amount of *à priori* knowledge available on the parameters of the stereo system; it can be identified in three cases [10]. First of all, if both intrinsic and extrinsic parameters are known, we can solve the reconstruction problem unambiguously by triangulation. Secondly, if only the intrinsic parameters are known, we can still solve the problem and, at the same time, estimate the extrinsic parameters of the system up to a scaling factor. Finally, if the pixel correspondences are the only information available, and neither the intrinsic nor the extrinsic parameters are known, we can still obtain a reconstructed environment using a global projective transformation. Table 2.2 shows the summary of 3-D reconstruction of the three cases.

Table 2.2 Summary of 3-D reconstruction three cases [10]

A priori knowledge	3-D Reconstruction from two views
Intrinsic and extrinsic parameters	Unambiguous (absolute coordinates)
Intrinsic parameters only	Up to an unknown scaling factor
No information on parameters	Up to an unknown projective transformation of the environment

From the above sections on stereovision problems, rectification techniques, and the corresponding search methods, we can reconstruct the 3-D scene following the following steps:

- (1) Detection of features (such as points or lines) in the two images;
- (2) For a given feature in the left image (say), perform a correspondence search.

(3) Calculation of depth using the disparity value.

The process is repeated using various features in the captured stereo images.

2.7 Summary

In this chapter, we have reviewed the available stereovision techniques, which include camera calibration, epipolar geometry constraints, rectification algorithms, stereo matching algorithms, and reconstruction algorithms. Based on the knowledge gained from the review, we shall employ the single-lens based prism based stereovision system, propose a geometry-based approach for virtual camera calibration and stereo images rectification, employ a cooperative optimization method to solve stereo correspondence problem, and implement triangulation for reconstruction in this thesis. The following chapters will describe these approaches in detail.

Chapter 3 Rectification of single-lens binocular stereovision system

Conventional stereovision systems employ two or more cameras to capture two or more different views of the same scene. The differences in positions of the correspondence points among these views are known as the disparities. Using the computed disparity values and also the geometric relations between each camera, the depth recovery and 3-D reconstruction of the captured scene becomes possible.

In this thesis, we used the single-lens bi-prism based stereovision system which employs only one camera. Every image captured using this system is divided into two sub-images, which are considered to be equivalent to two images captured using two virtual cameras generated by the bi-prism [21, 91-94]. However, the two virtual cameras, which form a virtual stereovision system, are non-coplanar due to the refraction through the bi-prism as shown in Figure 3.1 and 3.2.

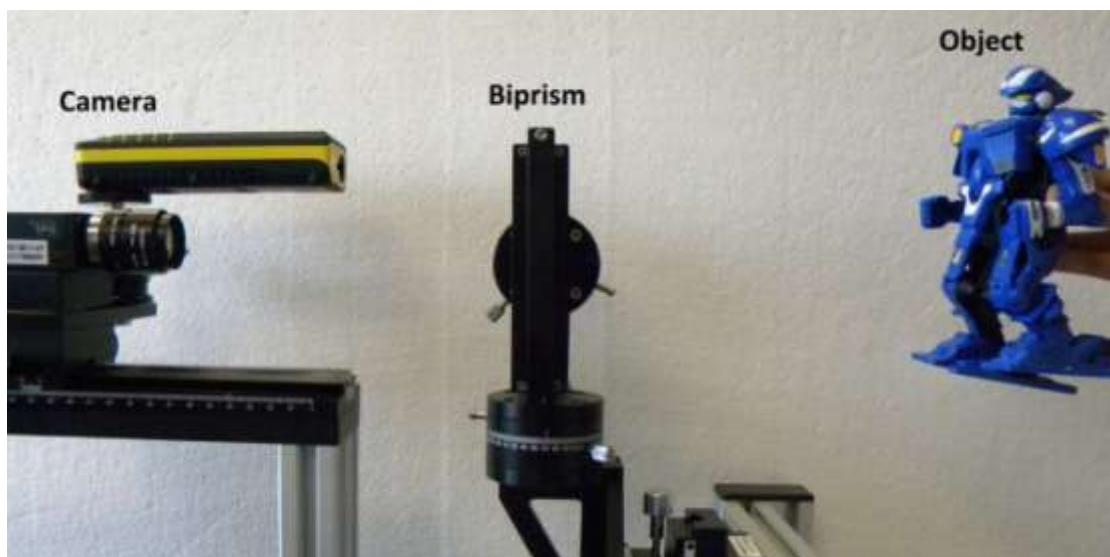


Figure 3.1 Single-lens based stereovision system using bi-prism

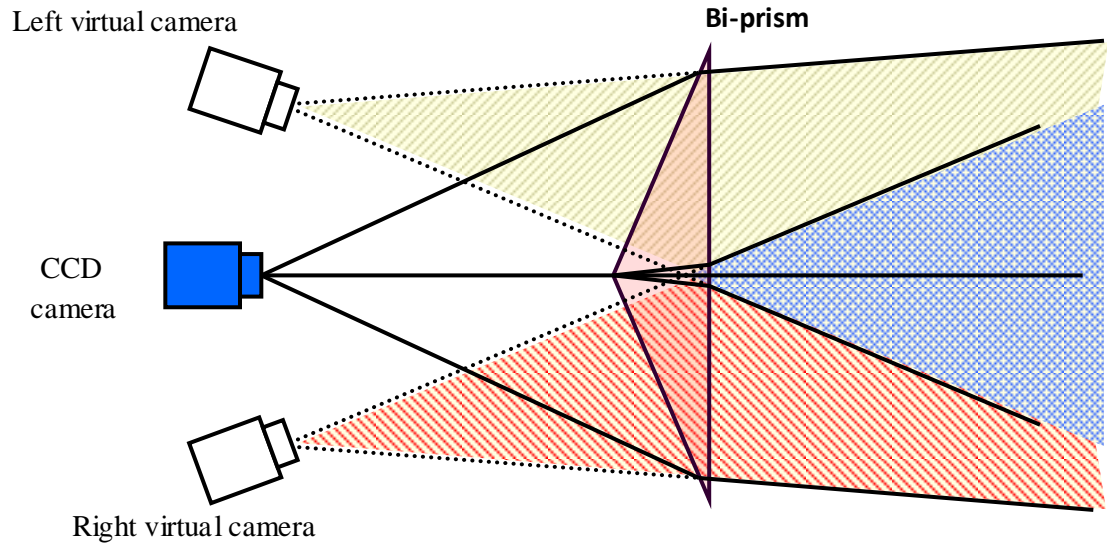


Figure 3.2 Single-lens stereovision using optical devices

In the determination of the said disparities, the basic problem is the search of correspondence points on the two stereo images. Epipolar constraint is then implemented so as to simplified the searching process. In stereovision systems, the setup is such that the two image planes are co-planar. The conjugate epipolar lines are therefore collinear (or near-collinear). This will then restrict the search for correspondence points to one dimension (section 2.5), hence making the computation to execute faster and to yield more accurate results. However, this assumption is not valid in the said system used in this thesis. To benefit from the said lower computation load in stereo correspondence search, in this chapter, we shall propose a rectification algorithm which transforms the non-coplanar virtual cameras to a set of coplanar virtual cameras.

A geometrical method is then used to calibrate the virtual cameras. Subsequently, the intrinsic and extrinsic parameters are used to rectify the epipolar lines of the image pairs which have been captured using this system. Part of the work in this chapter has been published in [123].

3.1 The background of stereo vision rectification

This section introduces the background of stereo vision rectification, which includes the pinhole camera-model, and the epipolar geometry after rectification. These principles will be applied in the next sections.

3.1.1 Camera model

A pinhole camera is modeled by its optical center C and its image plane. A 3D point P_w is projected into an image point m given by the intersection of image plane with the line containing C and P_w . The line containing C and orthogonal to the image plane is called the optical axis and its intersection with the image plane is the principal point (o_x, o_y) . The distance between C and the image plane is the focal length (Figure 3.3).

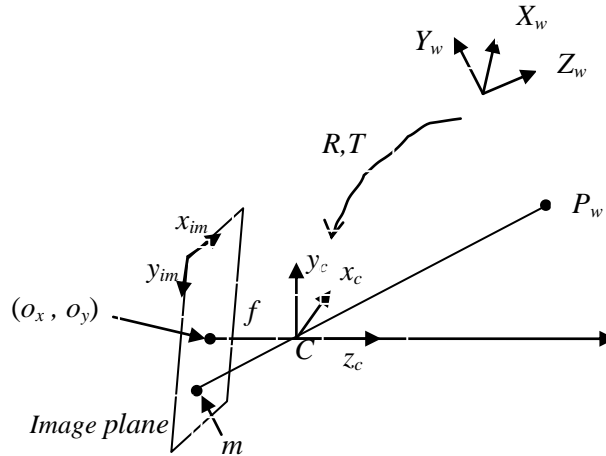


Figure 3.3 Pinhole camera model

Let $P_w = [x, y, z]^T$, the point in the 3-D space with respect to the world reference frame and $m = [u \ v]^T$, the point in the image plane (pixels) with respect to the image coordinate frame. The mapping from 3D coordinates to 2D coordinates is the perspective projection, which is represented by a linear transformation in homogenous coordinates. Let $\tilde{m} = [u \ v \ 1]^T$

and $\widetilde{P}_w = [x \ y \ z \ 1]^T$ be the homogeneous coordinates of m and P_w , respectively; then, the perspective transformation is given by the matrix P_{ppm} :

$$s\widetilde{m} = P_{ppm}\widetilde{P}_w \quad (3.1)$$

where s is a scale factor. The camera is therefore modeled by its *perspective projection matrix* (henceforth called *PPM*), which can be decomposed, into the product

$$P_{ppm} = M_{int}[R \ | \ T] \quad (3.2)$$

The matrix M_{int} is the matrix containing the intrinsic parameters and has the following form:

$$M_{int} = \begin{bmatrix} \frac{f}{s_x} & 0 & o_x \\ 0 & \frac{f}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f is the focal length, s_x and s_y are the effective size of the pixels (in millimeter) in the horizontal and vertical directions, respectively. The camera position and orientation (extrinsic parameters), are encoded by the 3×3 rotation matrix R and the translation vector T , which represent the rigid transformation that relate the camera reference frame to the world reference frame.

3.1.2 Rectification of image planes

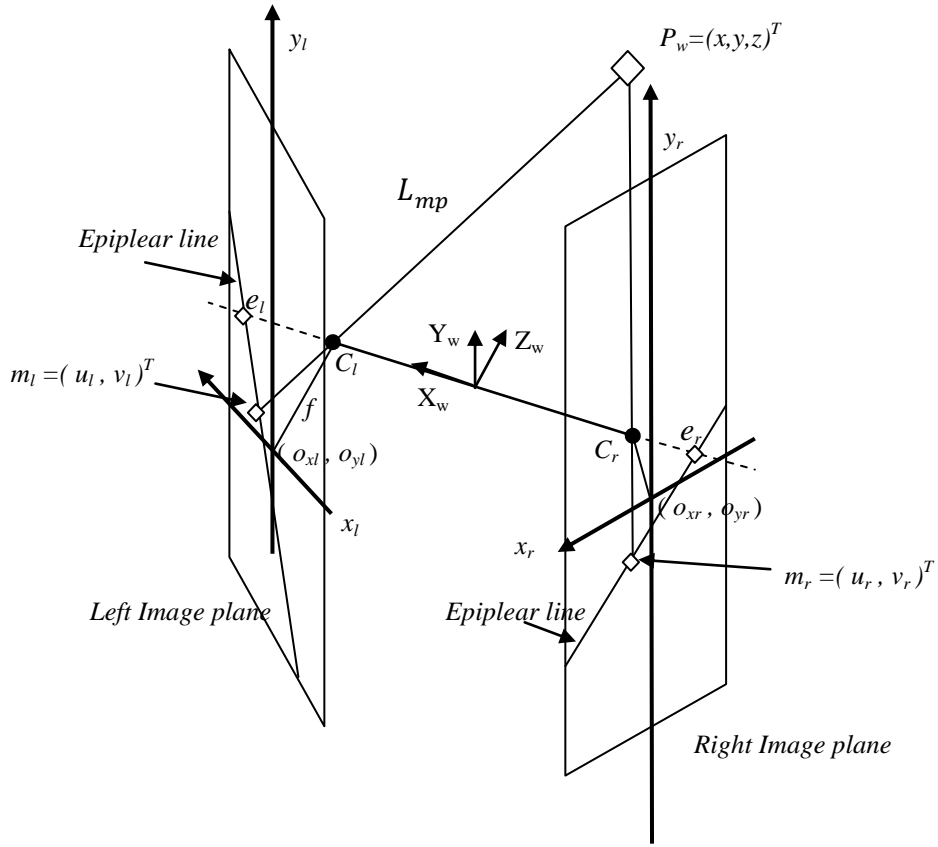


Figure 3.4 Epipolar geometry of two views

Let $Q = \begin{bmatrix} q_1^T & q_{14} \\ q_2^T & q_{24} \\ q_3^T & q_{34} \end{bmatrix}$, where $q_1 = \begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \end{bmatrix}$, $q_2 = \begin{bmatrix} q_{21} \\ q_{22} \\ q_{23} \end{bmatrix}$, $q_3 = \begin{bmatrix} q_{31} \\ q_{32} \\ q_{33} \end{bmatrix}$, be the *PPM* which is obtained

through camera calibration. By referring to Figure 3.4, the relationship between a point P_w in the world coordinates and its projection image point $m_l = [u_l, v_l]^T$ in the left image plane coordinates is expressed in the following equations (derived from Eq. (3.1))

$$\begin{cases} q_1^T P_w + q_{14} - u_l(q_3^T P_w + q_{34}) = 0 \\ q_2^T P_w + q_{24} - v_l(q_3^T P_w + q_{34}) = 0 \end{cases} \quad (3.3)$$

Eq. (3.3) can be rewritten as

$$\begin{cases} (q_1 - u_l q_3)^T P_w + q_{14} - q_{34} u_l = 0 \\ (q_2 - v_l q_3)^T P_w + q_{24} - q_{34} v_l = 0 \end{cases} \quad (3.4)$$

In Figure 3.4, The ray passing through the point P_w , m_l , and the optical center C_l is the line, L_{mp} , which is determined by the coordinates of the optical center C and P_w [125]. From Eq. (3.4), we observe that this equation is composed of two equations of planes [124].

The normal vectors of the two planes are $n_1 = (q_1 - u_l q_3)$ and $n_2 = (q_2 - v_l q_3)$, thus the direction of L_{mw} is $n = n_1 \times n_2 = [q_2 \times q_3 \ q_3 \times q_1 \ q_1 \times q_2] \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix}$. Therefore, L_{mp} can be defined as $P_w = C_l + \lambda n$, where $\lambda, \lambda \in \mathbb{R}$, is a real number correlated with P_w .

We shall now describe the fundamental principles of rectification. Figure 3.5 shows the geometry of a rectified stereovision setup. The discussion above will be extended here.

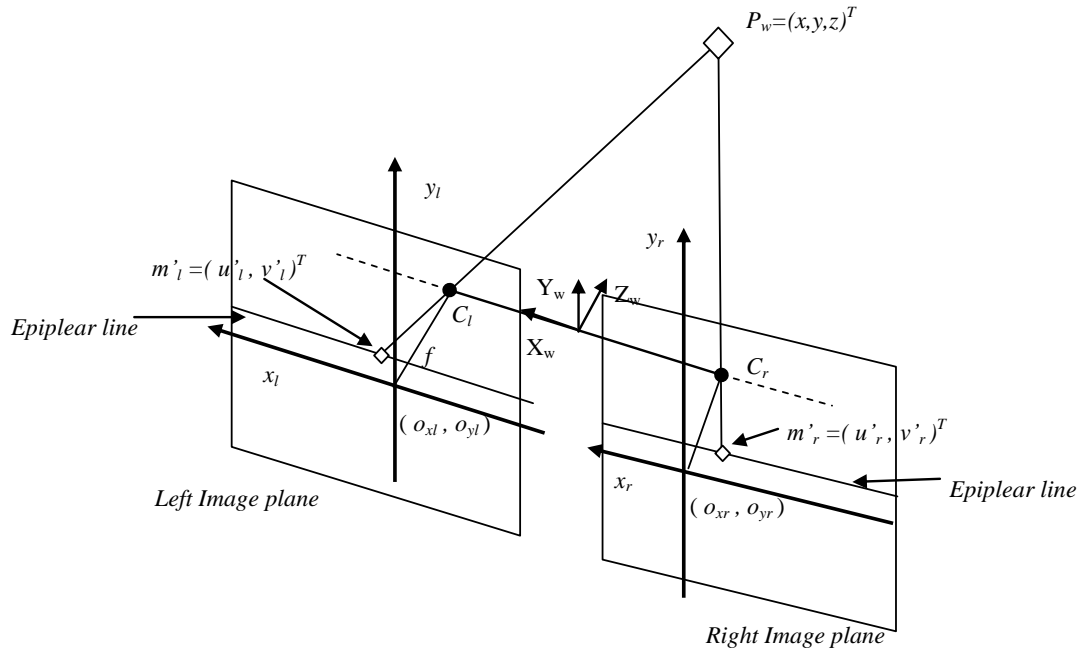


Figure 3.5 Rectified cameras. Image planes are coplanar and parallel to baseline

Figure 3.5 shows the geometry of a pair of rectified stereo images. After rectification the new perspective projection matrix, Q_{new} , can be written as

$$Q_{new} = \begin{bmatrix} m_1^T & m_{14} \\ m_2^T & m_{24} \\ m_3^T & m_{34} \end{bmatrix}, \text{ where } m_1 = \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \end{bmatrix}, m_2 = \begin{bmatrix} m_{21} \\ m_{22} \\ m_{23} \end{bmatrix}, m_3 = \begin{bmatrix} m_{31} \\ m_{32} \\ m_{33} \end{bmatrix}$$

Based on the same reasoning in obtaining Eq. (3.1), the perspective transformation is expressed as

$$s_{new} \begin{bmatrix} u'_l \\ v'_l \\ 1 \end{bmatrix} = Q_{new} \begin{bmatrix} P_w \\ 1 \end{bmatrix} = Q_{new} \begin{bmatrix} C_l + \lambda n \\ 1 \end{bmatrix} \quad (3.5)$$

where $\begin{bmatrix} u'_l \\ v'_l \\ 1 \end{bmatrix}$ is the pixel coordinates of the rectified left image point corresponding to P_w , and

s_{new} is a new scale factor in the new camera coordinate system. Since $Q_{new} \begin{bmatrix} C_l \\ 1 \end{bmatrix} = 0$, Eq.(3.5)

can be written as

$$s_{new} \begin{bmatrix} u'_l \\ v'_l \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} n,$$

Thus,

$$\begin{bmatrix} u'_l \\ v'_l \\ 1 \end{bmatrix} = \frac{\lambda}{s_{new}} \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} [q_2 \times q_3 \quad q_3 \times q_1 \quad q_1 \times q_2] \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} \quad (3.6)$$

It is clearly shown that the parameters involved in rectification are only in $\begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix}$. It can also be

written as $\begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} = M_{int} R$

which means the rectification matrix is related to the product of the intrinsic parameters matrix and the cameras pose matrix.

From Trucco [10], we have

$$p_r^T F p_l = 0 \quad (3.7)$$

$$F e_l = 0 = F^T e_r \quad (3.8)$$

where the fundamental matrix, F is a 3×3 matrix. p_l and p_r are the image points of in the left and right images, respectively. e_l and e_r are the epipoles in the left and right images, respectively (see Figure 3.3).

The rectification can be achieved by computing a pair of transformation matrices that map the epipoles to a point at infinity (Hartley's algorithm [42]). Therefore, the epipoles for a rectified pair of images are given as

$$e_{l\infty} = e_{r\infty} = [1 \ 0 \ 0]^T \quad (3.9)$$

and the fundamental matrix has the form of

$$F_\infty = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} . \quad (3.10)$$

3.2 Rectification of single-lens binocular stereovision system using geometrical approach

In this section, we propose a rectification approach and describe it in several subsections which include the computation of the virtual cameras extrinsic parameters, and the formation of the rectification transformation matrix on the image pairs.

The steps of the proposed rectification algorithm on single-lens prism based stereovision system are shown below:

- 1) Determine the virtual cameras' projection matrix based on geometrical analysis (section 3.2.1).
- 2) Build the rotation matrix of the new camera coordinates of the two virtual cameras (Eq.(3.36)).
- 3) Compute rectification transformation matrix which move the epipoles to infinity (Eq.(3.37)).
- 4) Apply the rectification transformation matrix on the image pairs.

3.2.1 Computation of the virtual cameras' projection matrix

In this section, we present the proposed geometrical approach to determine the extrinsic parameters of the single-lens stereovision system using bi-prism. Conventionally, the extrinsic parameters are obtained through the camera calibration process. However, for this system, we apply the principles of optics and employ a geometry-based method to compute the extrinsic parameters.

1) The basic idea of the proposed geometrical approach

Figure 3.1 shows our stereovision setup, with a bi-prism placed in front of a camera; the output image captured by the camera contains two sub-images of the same scene (left and right of the image planes) behind the prism. This image is equivalent to images captured using two camera system (called virtual cameras) with specific position and orientation. The formation of the images by the two virtual cameras is illustrated in Figure 3.2.

To determine the position and orientation of the virtual camera's image plane and its optical centre, the extrinsic parameters of the system are determined under the following conditions and assumptions:

- a) The bi-prism is symmetrical with respect to its apex line.
- b) The back plane of the prism is parallel to the image plane of the real camera.
- c) The centre point of the image plane is located on the Z_w -axis which means the projection of the bi-prism apex will bisect the image plane into half equally.
- d) The real image plane of camera has consistent properties, such as pixels size, distortion free and focal length.

Assuming the four conditions above are satisfied, the steps of our proposed geometrical approach to find the rectification transformation matrices are described as follows (refer to Figures 3.6 and 3.7).

Note that we shall describe only the steps involved in the determination of the extrinsic parameters of the left virtual camera plane. That of the right virtual camera plane can be determined in the similar way:

- (1) Arbitrary image points from the right half of the real camera image plane are chosen (see Figure 3.7, P_a , P_b and P_c). Based on the pin-hole camera principle, the projection rays of these image points will pass through the optical centre of the real camera and are refracted at the left plane of the prism.
- (2) The ray from each of the image points are refracted to reach the scene behind the prism.
Taking the image point P_a as illustration, Ray P_aA , after two refraction, will exit the prism through Ray 3. By back extending Ray 3, it will pass through the optical centre of the left virtual camera, based on the pin-hole camera principles. Therefore, the back

extended rays of any individual image points must intersect each others at the same point, which is the optical centre of the left virtual camera. The back extended rays will all fall on the image plane of the left virtual camera.

- (3) From the above, the intersection of the back-extended rays gives the position of the left virtual camera's optical centre with respect to the world coordinate frame. This will yield the translational relationship between the left virtual camera and the optical centre of the real camera (see Figure 3.10).
- (4) To determine the orientation of the virtual cameras, we consider the centre point of the image plane of the real camera image plane, P_c (see Figure 3.7). The ray from this point will go along the Z_w -axis of the world coordinate frame and will pass through the optical centre of the real camera. This ray will be refracted at both faces of the prism. In the case of the ray refracted on the left hand face of the prism (see Figure 3.6, plane Π_1), the ray leaving the end face of the prism is indicated as Ray 3C in figure 3.7. The back extended ray of Ray 3C will be the optical axis of the left virtual camera.
- (5) In Figure 3.7, the angle, β , represents the angle of inclination of the left virtual camera image plane with respect to the X_w -axis of the world coordinate frame. This angle will be useful in determining the rotational relationship between the left virtual cameras image plane and the real image plane.
- (6) Step (1) to (4) allows the determination of the optical centre and the optical axis of the left virtual camera. Together with Step (5), we will be able to determine the extrinsic parameters of the left virtual cameras.

2) Determination of the virtual cameras extrinsic parameters based on geometrical analysis

Camera calibration is a process to obtain the intrinsic and extrinsic parameters of the camera. The intrinsic parameters, such as the focal length, the lens distortion, and the scaling factors can be obtained from specifications of the hardware employed in the system. The extrinsic parameters include the translational and orientation properties of the camera coordinates frame with respect to a pre-defined world coordinate system. In a stereovision system, the extrinsic parameters are essential (the relative position and orientation between the two or more cameras) in 3D reconstruction and depth recovery.

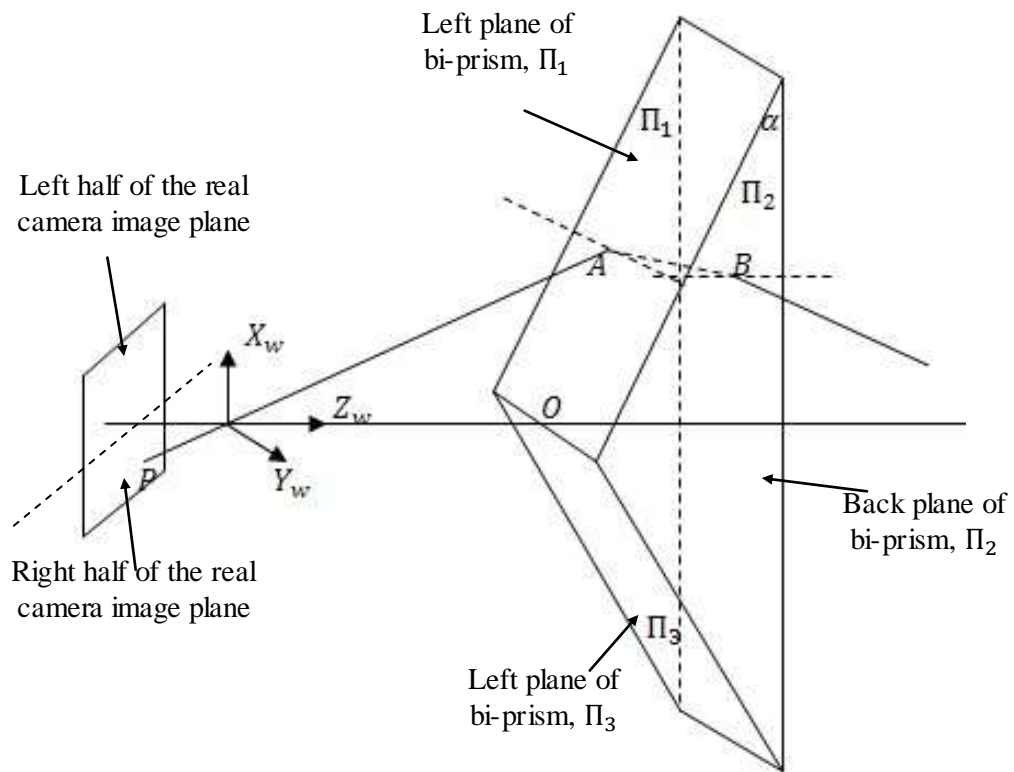


Figure 3.6 Geometry of single-lens bi-prism based stereovision system (3D)

For our single-lens prism based stereovision system, we propose a simple technique to estimate the extrinsic parameters of the cameras based on geometrical analysis of ray sketching.

To determine the extrinsic properties of the virtual cameras, we define the following parameters (refer to Table 3.1 and to Figure 3.7):

Table 3.1 The parameters of single-lens stereovision using biprism

Parameters	Definition
λ	distance between the two virtual camera optical centers or baseline
β	the rotational angle of the virtual camera with respect to the real camera. It should be the same for the two virtual cameras.
f	is the focal length of the real and virtual cameras
α	the corner angle of the bi-prism
n	the refractive index of the bi-prism glass material
T_o	the distance of the real camera optical centre from the apex of the bi-prism
T	the distance between the optical center and the bi-prism's back plane
I	two dimensional matrix representing the pixels of the real camera image plane

Figure 3.7 is the top view of the setup schematically, in which the world coordinate frame is given by (X_w, Y_w, Z_w) and that of the left virtual camera image plane is represented by (x_L, y_L, z_L) . The real camera optical axis Z_w can be considered as the demarcation line that bisects the setup into two halves, left and right. In Figure 3.7, the part that is above Z_w is the left half and that below is the right half.

We shall consider only the left virtual camera as shown in Figure 3.7. Taking an arbitrary point on the right image plane of the real camera, $P_a(x_a, y_a, -f)$, a series of angles ϕ_1, ϕ_2, ϕ_1' , and ϕ_2' , are formed between the incident ray and the normals of the two faces of the prism (see Figure 3.7). Ray 1 is the incident ray from point P_a , Ray 2 is the refracted ray in the prism and Ray 3 is the second refracted ray which exits from the back plane of the prism. Point A is the intersection between Ray 1 and the bi-prism half plane while point B is the

intersection between Ray 2 and the back plane of the bi-prism. According to the law of refraction, the following expression can be obtained:

$$n = \frac{\sin \phi_1}{\sin \phi'_1} = \frac{\sin \phi_2}{\sin \phi'_2} \quad (3.11)$$

(1) Equation of Ray 1

Given a point P_a in world coordinate system, we can obtain the line equation for Ray 1 as:

$$\frac{X}{x_a} = \frac{Y}{y_a} = \frac{Z}{-f} \quad (3.12)$$

(2) Equation of Ray 2

Let $n_{\Pi_1} = [n_x, n_y, n_z]$, be the normal vector of plane Π_1 (see Figure 3.6). The plane equation of Π_1 is expressed as

$$n_x(X - X_0) + n_y(Y - Y_0) + n_z(Z - Z_0) = 0 \quad (3.13)$$

where (X_0, Y_0, Z_0) is a point on the plane Π_1 .

By referring to Figures 3.6 and 3.7, we can establish the three following conditions for the plane Π_1 :

Condition 1, apex point $O(0,0, T_0)$ lies on the plane Π_1 ;

Condition 2, apex of the bi-prism is parallel to the Y_w -axis based on the setup specification (see Figure 3.6);

Condition 3, angle between planes Π_1 and Π_3 is the corner angle of the prism which is α .

Considering the above three conditions, we obtain

$$\begin{cases} n_x X + n_y Y + n_z (Z - T_0) = 0 \\ n_y = 0 \\ \frac{n_z}{\sqrt{(n_x^2 + n_y^2 + n_z^2)}} = \cos(\alpha) \end{cases} \quad (3.14)$$

By solving Eq. (3.14), the equation of Π_1 is

$$X \sin(\alpha) - Z \cos(\alpha) + T_0 \cos(\alpha) = 0 \quad (3.15)$$

Then, the normal vector of Π_1 is

$$n_{\Pi_1} = [\sin(\alpha), 0, -\cos(\alpha)]$$

n_{Π_1} is also a unit direction vector.

The coordinates of point A can be obtained by considering the intersection between the left bi-prism half plane and Ray 1 represented by Eq. (3.12):

$$A = [x_A, y_A, z_A] = \left[\frac{-T_0 x_a \cos(\alpha)}{f \cos(\alpha) + x_a \sin(\alpha)}, \frac{-T_0 y_a \cos(\alpha)}{f \cos(\alpha) + x_a \sin(\alpha)}, \frac{T_0 f \cos(\alpha)}{f \cos(\alpha) + x_a \sin(\alpha)} \right]$$

ϕ_1 is the angle between Ray 1 and the bi-prism half plane's normal vector. Considering the unit direction vector of Ray 1 and unit normal vector of plane Π_1 , we form an expression for ϕ_1 as shown in Eq.(3.16),

$$\cos \phi_1 = \frac{f \cos(\alpha) + x_a \sin(\alpha)}{\sqrt{x_a^2 + y_a^2 + f^2}} \quad (3.16)$$

From Eq. (3.11) and Eq. (3.16), we obtain

$$\phi'_1 = \sin^{-1}\left(\frac{\sin \phi_1}{n}\right) \quad (3.17)$$

Next, we shall determine the direction vector of Ray 2 using parallelogram rule.

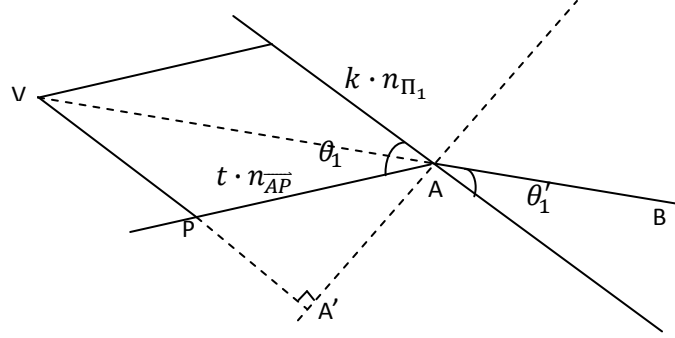


Figure 3.8 The relationship of direction vector of AB and normal vector of plane Π_1

Referring to Figure 3.8, the unit direction vector Ray 2 is expressed in terms of the unit direction vector n_{Π_1} and the unit direction vector n_{AP} ,

$$n_{AB} = -(k \cdot n_{\Pi_1} + t \cdot n_{AP}) \quad (3.18)$$

Then, we obtain the equation from Figure 3.8 as follows:

$$\begin{cases} |VA'| = |n_{VA}| \cdot \cos \theta'_1 \\ |VA'| = |VP| + |PA'| = |k \cdot n_{AO_2B}| + |PA'| \\ |PA'| = |m \cdot n_{\Pi_1}| \\ |AA'| = |n_{VA}| \cdot \sin \theta'_1 = |t \cdot n_{AP}| \cdot \sin \theta_1 \\ |VA| = |n_{BA}| \end{cases} \quad (3.19)$$

where k, t , and m are scalar.

From Eq.(3.19), the solution of k, t and m are expressed as follows:

$$\begin{cases} k = \frac{\sin\theta_1'}{\sin\theta_1} \\ m = \frac{\sin\theta_1' \cdot \cos\theta_1}{\sin\theta_1} \\ t = \frac{\sin(\theta_1 - \theta_1')}{\sin\theta_1} \end{cases} \quad (3.20)$$

The unit direction vector of AB , $n_{\overline{AB}} = [x_{\overline{AB}}, y_{\overline{AB}}, z_{\overline{AB}}]$ is described by

$$n_{\overline{AB}} = -\left(\frac{\sin\theta_1'}{\sin\theta_1} \cdot n_{\Pi_1} + \frac{\sin(\theta_1 - \theta_1')}{\sin\theta_1} \cdot n_{\overline{PM}}\right) \quad (3.21)$$

Thus, the equation of Ray 2 can be found from point A and $n_{\overline{AB}}$.

$$\frac{X - x_A}{x_{\overline{AB}}} = \frac{Y - y_A}{y_{\overline{AB}}} = \frac{Z - z_A}{z_{\overline{AB}}} \quad (3.22)$$

Point B can be found from the intersection between Ray 2 and the bi-prism back plane ($Z = T$) which is known.

$$\text{So, the coordinates of } B = \left[\frac{(T - z_A)x_{\overline{AB}} + x_{AZ}\overline{AB}}{z_{\overline{AB}}}, \frac{(T - z_A)y_{\overline{AB}} + y_{AZ}\overline{AB}}{z_{\overline{AB}}}, T \right] \quad (3.24)$$

Here, ϕ_2' is the angle between Ray 2 and bi-prism back plane's normal vector,

$$\cos\phi_2' = n_{\overline{AB}} \cdot n_{\overline{Zw}} \quad (3.24)$$

From Eq.(3.11) and Eq.(3.24), we obtain:

$$\phi_2 = \sin^{-1}(n \times \sin\phi_2') \quad (3.25)$$

(3) Equation of Ray 3

In order to determine the equation of Ray 3, we must first determine the direction vector of Ray 3 which is $n_{\overrightarrow{ray3}} = [x_{\overrightarrow{ray3}}, y_{\overrightarrow{ray3}}, z_{\overrightarrow{ray3}}]$. We also employ the parallelogram rule to obtain $n_{\overrightarrow{ray3}}$.

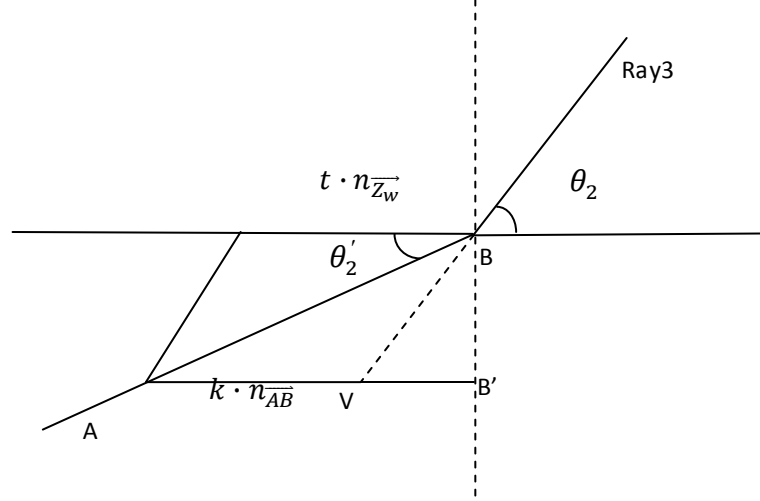


Figure 3.9 The relationship of direction vector of AB and normal vector of plane Π_3

The $n_{\overrightarrow{ray3}}$ can be expressed by

$$n_{\overrightarrow{ray3}} = k \cdot n_{\overrightarrow{AB}} - t \cdot n_{\overrightarrow{zw}} \quad (3.26)$$

Referring to Figure 3.9, we obtain the following equations:

$$\begin{cases} |BB'| = |k \cdot n_{\overrightarrow{AB}}| \cdot \sin \theta'_2 \\ |AB'| = |AV| + |VB'| = |t \cdot n_{\overrightarrow{z}}| + |n_{\overrightarrow{VB}}| \cdot \cos \theta_2 \\ |BB'| = |n_{\overrightarrow{VB}}| \cdot \sin \theta_2 \\ |AB'| = |k \cdot n_{\overrightarrow{AB}}| \cdot \cos \theta'_2 \\ |VB| = |n_{\overrightarrow{ray3}}| \end{cases} \quad (3.27)$$

From Eq. (3.27), the solution of k, t are obtained as follows:

$$\begin{cases} k = \frac{\sin \theta_2}{\sin \theta'_2} \\ t = \frac{\sin(\theta_2 - \theta'_2)}{\sin \theta'_2} \end{cases} \quad (3.28)$$

Thus, the unit direction vector n_{ray3} is

$$n_{ray3} = \frac{\sin\theta_2}{\sin\theta'_2} \cdot n_{AB} - \frac{\sin(\theta_2 - \theta'_2)}{\sin\theta'_2} \cdot n_{zw} \quad (3.29)$$

The equation of Ray 3 can be found from point B and n_{ray3} in the same way as above.

$$\frac{X - x_B}{x_{ray3}} = \frac{Y - y_B}{y_{ray3}} = \frac{Z - z_B}{z_{ray3}} \quad (3.30)$$

Based on pin-hole camera model, the back extension of Ray 3 will pass through the optical centre and reach the image plane of the left virtual camera which is shown in Figure 3.7.

(4) Determination of the optical centre of the virtual camera

By taking another arbitrary image point (P_b on the right half of the real camera image plane and repeat the same procedure as above, the back extension of Ray 3B will go through the optical centre and reach the image plane of the left virtual camera like in the case of P_a presented above. The optical centre of the virtual camera ($OC_L = [OC_{Lx}, OC_{Ly}, OC_{Lz}]^T$) can be recovered by computing the intersecting of the back extension ray of Ray 3 and Ray 3B. In order to obtain a more accurate optical centre of the virtual camera, we repeat the above steps with more points and compute the average of the results. The average coordinates are taken as the optical centre of the virtual camera.

Theoretically, the back extension ray of Ray 3 and Ray 3B should intersect, however due to digitization errors, this may not be the case. Therefore an approximation of the coordinates of OC_L may be necessary if this situation occurs.

For two straight lines in 3D that do not intersect and are not parallel to each other, the unique shortest distance between Ray 3 and Ray 3B should be considered. If the shortest distance

between the two lines is given by EF, and that points E and F are on the two straight lines. The mid-point of EF is taken to be the optical center of the virtual camera (refer to Appendix A).

(5) Determination of the rotation angle β

As the size of the camera sensor (I) is known, we shall now consider the centre of the real image plane (P_c). It lies on the Z_w -axis of the world coordinate frame (also the optical axis of the real camera). Based on the pin hole camera model, the ray originating from P_c will pass through the optical centre of the real camera and it will form an incident ray (Ray 1C) which is coincident with the Z_w -axis as shown in Figure 3.7. This ray will be refracted (Ray 2C) at the apex of the bi-prism and refracted again (Ray 3C) at the back plane of the bi-prism. According to the law of refraction, we have

$$n = \frac{\sin \phi_3}{\sin \phi'_3} = \frac{\sin \phi_4}{\sin \phi'_4} \quad (3.31)$$

Referring to Figure 3.7, we can obtain the following relationship:

$$\phi_3 = \alpha$$

$$\phi'_4 = \alpha - \phi'_3 \quad (3.32)$$

$$\phi_4 = \sin^{-1}(n \sin \phi'_4) \quad (3.33)$$

The coordinates of C , the intersection of Ray 2C and the bi-prism back plane can be computed, and the equation for Ray 3C can be recovered from point C and ϕ_4 . The back extension of Ray 3C will again pass through the optical centre of the left virtual camera and this back extended ray will form the optical axis of the left virtual camera.

β , the angle between the left virtual camera image plane with the X_wY_w plane of the world coordinate system, is also the orientation of the virtual camera with respect to the world coordinate system (see Figure 3.7). Using simple trigonometry, and the principle of similar triangle, we can obtain from Figure 3.7

$$\beta = \phi_4 \quad (3.34)$$

(6) Determination of the Rotational Matrix and Translation Vector, R_L and T_L

From the above development, we can derive the rotational matrix R_L and the translation vector T_L , which are the rotational and translational transformation of an image point on the real image plane to the left virtual camera image plane.

$$\text{Rotational matrix, } R_L = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$\text{Translation vector, } T_L = \begin{bmatrix} OC_{Lx} \\ OC_{Ly} \\ OC_{Lz} \end{bmatrix}$$

The camera intrinsic matrix M_L is known for our stereovision system.

$$\text{Camera intrinsic matrix, } M_L = \begin{bmatrix} -f_x & 0 & C_x \\ 0 & -f_y & C_y \\ 0 & 0 & 1 \end{bmatrix}$$

where:

f_x , focal length of the virtual camera in the x-direction in terms of pixel size

f_y , focal length of the virtual camera in the y-direction in terms of pixel size

C_x , coordinate of the virtual camera image center in x-direction

C_y , coordinate of the virtual camera image center in y-direction

The projection matrix which relate the 3D coordinates to 2D coordinate of the virtual camera image plane is given by [10] :

$$P_L = M_L[R_L|T_L], \quad P_R = M_R[R_R|T_R] \quad (3.35)$$

where R_L and R_R are the rotation matrices of the left and right virtual image planes, respectively. T_L and T_R are the translation vectors of the left and right virtual image plane, respectively. M_L and M_R are the intrinsic matrices of the left and the right virtual camera, respectively. P_L and P_R are the perspective projection of left and right virtual camera, respectively.

3.2.2 Rectification Algorithm

Based on the epipolar geometry [10], given a point p'_{l1} on the left virtual camera, Fp'_{l1}^T is the epipolar line of p'_{r1} on the right virtual camera, where F is the fundamental matrix of the two virtual cameras. By computing the epipolar line for another point p'_{l2} on the left virtual camera, the intersection of the two epipolar lines will give us the epipole of the virtual camera.

Next, we shall form the rectification transformation matrix which projects the image points on both the left and right image planes of the virtual cameras to two coplanar planes (Figure 3.10). This transformation will make the epipolar lines parallel to the a horizontal scan-line and move the epipoles of the new image planes of the virtual cameras to infinity as shown in Figure 3.5 and Eq. (3.9).

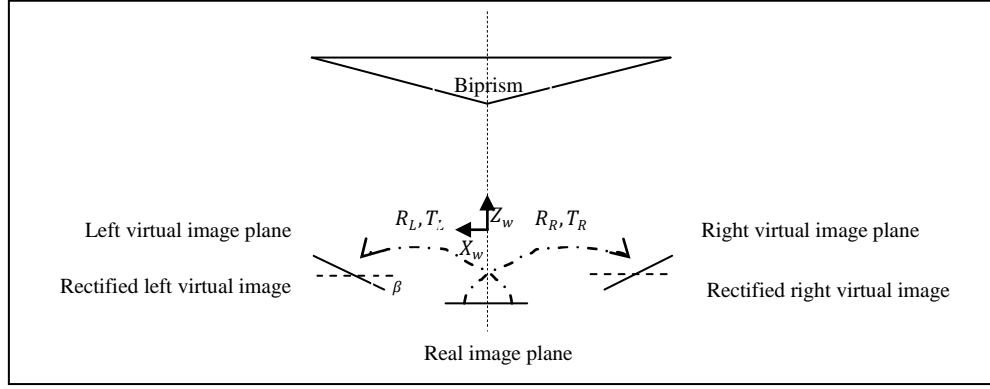


Figure 3.10 Rectification of virtual image planes

The new virtual camera coordinates are defined as

$$R_{rect} = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix} \quad (3.36)$$

Based on our current setup, we define

r_1 as the new X -axis of the virtual camera which is chosen to be parallel to the baseline:

$$r_1 = \frac{OC_{Lx} - OC_{Rx}}{\|OC_{Lx} - OC_{Rx}\|}$$

r_2 as the new Y -axis of the virtual camera which is the same as the original Y_w -axis

r_3 is the new Z -axis of virtual camera which is the cross product of r_1 and r_2

The final rectification transformation matrices of the two virtual image plane are:

$$R_l = R_L R_{rect}, \quad R_r = R_R R_{rect} \quad (3.37)$$

Following the above steps, we will obtain the rectification image coordinates.

Rectification algorithm

1. Compute the R_L and T_L (transform the right half plane of real image plane to the left virtual image plane);
 2. The relationship of camera point on the right half plane of real camera, $p_r = [x, y, -f]$, with the camera point on the left-virtual camera, p_{lv} ,

$$p_{lv} = R_L(p_r - T_L)$$
 3. Construct the matrix R_{rect} ;
 4. Set $R_l = R_L R_{rect}$ and $R_r = R_R R_{rect}$
 5. For each of the left-virtual camera point p_{lv} , compute

$$R_l p_{lv} = [x', y', z']$$
and the coordinates of the corresponding rectified point, p'_l as

$$p'_l = \frac{f}{z'} [x', y', z']$$
 6. Compute each rectified image coordinator of the left virtual camera

$$p_{lim} = M_L p'_l$$
 7. Repeat the previous step for the left half plane of the real camera to compute the rectified image coordinator of the right virtual camera.
-

After applying the rectification algorithm on the virtual image planes, the perspective projection matrix of the left and right virtual cameras becomes:

$$P'_L = M_L [R'_L | T_L], \quad P'_R = M_R [R'_R | T_R] \quad (3.38)$$

where R'_L and R'_R are the 3×3 identity matrix.

3.3 Experimental results and discussion

Figure 3.1 shows our single-lens prism based stereovision system setup used in this work. Four images were captured with bi-prisms of different prism angles and they are in Figure 3.11 (a) - 3.14 (a). They are respectively the images of “robot”, “soap bottle”, “cif”, and “pets”. Each image consists of two halves of the same object and background. This is the results of using our stereovision system mentioned above. The two half images are considered to be equivalent to two images captured using two virtual cameras. The epipolar lines of the captured images are also plotted and shown in the same figures (Figure 3.11 (a) - 3.14 (a)). Rectification was then applied on the captured images, and the epipolar lines are plotted on

Figures 3.11(b) to 3.14(b). The epipolar lines of the rectified images appear to be along a horizontal scan line of the images. The experiment is designed to test the performance of the proposed rectification algorithm. The steps involved are:

- The first step is to ensure sure that the camera and the bi-prism are positioned correctly in order to reduce errors due to the hardware setup.
- Secondly, then, the virtual cameras are calibrated using the proposed geometrical approach (section 3.2.1).
- Thirdly, image of the target object and the background are captured by the stereovision system.
- The final step is to apply the rectification transformation matrix on the images.

The values of the parameters for the single-lens prism based stereovision system used in the experiment are shown in Table 3.2.

Table 3.2 The values of parameters for bi-prism used in the experiment

α	n	f, f_x, f_y	T	T_o	C_x	C_y
$6.4^\circ, 20^\circ, 45^\circ, 10^\circ$	1.5	25 mm	215.6 mm	210 mm	384	521

For all the four images used in the experiment, we carried out the following steps after ensuring the setting of the hardware are accurately done and the calibration of the system has been properly carried out:

In each case, three points are selected on the left image and their correspondence epipolar lines are plotted on the right image. Figures 3.11(a) - 3.14 (a) show the said epipolar lines in the four experiments. After implementing the rectification algorithm, the three points are rectified and their correspondence epipolar lines are also rectified. Figures 3.11(b) - 3.14 (b) show that the original epipolar lines are rectified and appear as horizontal lines (parallel or

along a scan line of the image) in the rectified images. This demonstrates that our algorithm is correct.

The projection matrices of two virtual cameras (which are generated using the bi-prism with angle of 6.4°) were calculated using the geometrical approach (section 3.2.1):

$$P_{OL} = \begin{bmatrix} -0.5400 \times 10^4 & 0 & -0.0090 \times 10^4 & 7.5880 \times 10^4 \\ -0.0071 \times 10^4 & -0.5376 \times 10^4 & 0.0632 \times 10^4 & 0.4452 \times 10^4 \\ 0 & 0 & 0.0001 \times 10^4 & 0.0007 \times 10^4 \end{bmatrix}$$

$$P_{OR} = \begin{bmatrix} -0.5286 \times 10^4 & 0 & 0.1108 \times 10^4 & -6.8712 \times 10^4 \\ 0.0071 \times 10^4 & -0.5376 \times 10^4 & 0.0632 \times 10^4 & 0.4452 \times 10^4 \\ 0 & 0 & 0.0001 \times 10^4 & 0.0007 \times 10^4 \end{bmatrix}$$

where P_{OL} and P_{OR} are the projection matrices of the left and right virtual image plane before rectification, respectively.

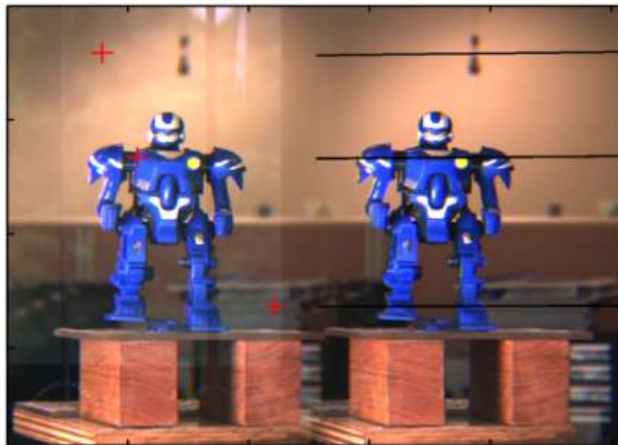
After the implementation of the rectification algorithm, the image planes of two virtual cameras become coplanar (R'_L and R'_R become identity matrices). The new projection matrices of two coplanar virtual image planes are as follow:

$$P_{NL} = M_L[R'_L T_L] = \begin{bmatrix} -0.5376 \times 10^4 & 0 & 0.1116 \times 10^4 & 7.5550 \times 10^4 \\ 0 & -0.5376 \times 10^4 & 0.0632 \times 10^4 & 0.3477 \times 10^4 \\ 0 & 0 & 0.0001 \times 10^4 & 0.0005 \times 10^4 \end{bmatrix}$$

$$P_{NR} = M_R[R'_R T_R] = \begin{bmatrix} -0.5376 \times 10^4 & 0 & 0.1117 \times 10^4 & -6.9944 \times 10^4 \\ 0 & -0.5376 \times 10^4 & 0.0632 \times 10^4 & 0.3477 \times 10^4 \\ 0 & 0 & 0.0001 \times 10^4 & 0.0005 \times 10^4 \end{bmatrix}$$

where P_{NL} and P_{NR} are the projection matrices of the left and right virtual image plane after rectification, and other parameters are defined in section 3.2.2.

Left image and Right image



(a)

Rectified left image and right image



(b)

Figure 3.11 $\alpha = 6.4^\circ$, “robot” image pair (a) and rectified image pair (b)

The algorithm in section 3.2.1 is repeated with the bi-prism angles of 10° , 20° and 45° . For simplicity, we only show the final results of the images (Figure 3.12-3.14)).

Left image and Right image



(a)

Rectified left image and right image



(b)

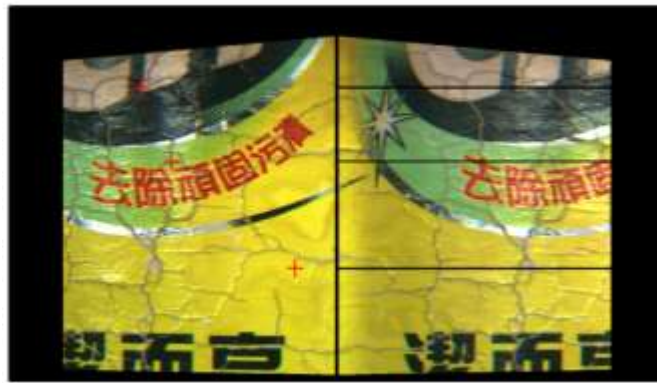
Figure 3.12 $\alpha = 20^\circ$, “soap bottle” image pair (a) and rectified pair (b)

Left image and Right image



(a)

Rectified left image and right image



(b)

Figure 3.13 $\alpha = 45^\circ$ "cif" image pair (a) and rectified pair (b)

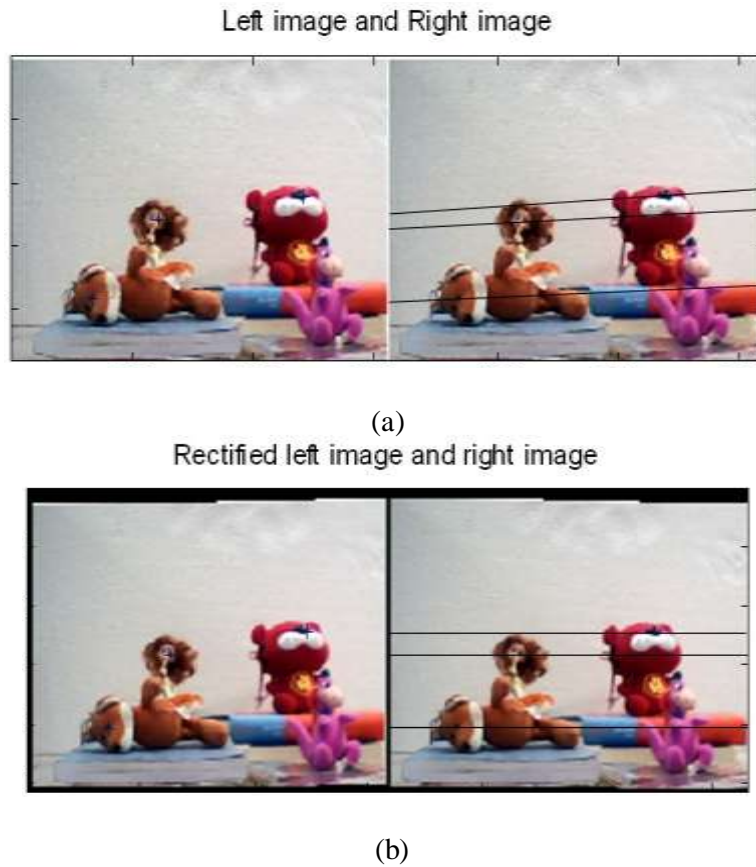


Figure 3.14 $\alpha = 10^\circ$, ‘Pet’ image pair (a) and rectified pair (b)

We have also conducted experiment to assess the performance of our stereo-correspondence method, or more precisely the geometrical approach. We also make comparison between the results of our method and the conventional calibration method (Zhang’s method) [29].

Table 3.4 shows the results of stereo correspondence using the conventional calibration method [29] and geometrical approach.

The following Table 3.3 gives the descriptions of the columns in Table 3.4. Note that the Euclidean distances calculated in column 6 and 7 indicate the errors in the determination of the correspondence points using the Geometrical approach and the conventional calibration method, respectively.

3.3 The descriptions of the columns in Table 3.4

Column Number	Description
1	Label of test point
2	Coordinates of the point in the left image
3	Correspondence point in the right image obtained using geometrical approach
4	Coordinates of the correspondence point obtained by direct measurement
5	Correspondence point in the right image obtained using conventional calibration method.
6	Euclidean distance between the correspondence points determined by geometrical approach (Column 3) and direct measurement (Column 4)
7	Euclidean distance between the correspondence points determined by conventional calibration method (Column 5) and direct measurement (Column 4)

Table 3.4 shows the numerical results in Column 6 are consistently lower than that in Column 7. The average values of the two columns are 6.7784 and 10.7526, respectively. This observation shows that our Geometrical approach is better than the conventional calibration method in searching for the correspondence point. The reason is being that the extrinsic parameters of virtual camera obtained by the geometrical approach are more accurate than those obtained by the conventional calibration method. In addition, the geometrical approach is specially designed to rectify the virtual cameras of the single-lens stereovision system. This also further verifies that our proposed geometrical approach is comparatively better in rectifying the virtual cameras.

Table 3.4 Results of conventional calibration method and geometrical method for obtaining stereo correspondence

points	Pixel coordinate (in the right half plane of real image plane)	Correspondence coordinate by Geometry Approach	Actual correspondence coordinate	Correspondence coordinate by Calibration Approach	Distance (in number of pixels) (Geometry)	Distance (in number of pixels)(Calibr ation)
Point 1	(58, 90)	(576, 89)	(573, 86)	(581, 92)	4.2426	10
Point 2	(100, 120)	(612, 126)	(617, 125)	(621, 134)	5.099	9.8489
Point 3	(157, 83)	(665, 88)	(670, 86)	(660, 84)	5.3851	10.1980
Point 4	(180, 200)	(694, 195)	(698, 187)	(702, 198)	8.9442	11.7047
Point 5	(216, 265)	(726, 261)	(732, 263)	(735, 256)	6.3246	7.6158
Point 6	(300, 350)	(806, 351)	(810, 354)	(815, 359)	5.000	7.0710
Point 7	(443, 526)	(950, 528)	(955, 521)	(961, 508)	8.6023	14.3178
Point 8	(480, 550)	(998, 537)	(991, 545)	(1004, 553)	10.6301	15.2643
AVG					6.7784	10.7526

3.4 Summary

The objectives of the rectification algorithm are to simplify the search of correspondence points and increase the speed of the stereo matching algorithm. We propose a simple geometrical ray sketching approach to compute the projection transformation matrix and the rectification transformation matrix to rectify the virtual cameras generated using the single-lens prism based stereovision system. Furthermore, the parallelogram rule and refraction rule are employed to determine the sketch ray functions can be easily used to obtain the desired rays and also reduce the computational error. The experimental result verifies the accuracy of the proposed approach. In the next chapter, we will discuss the rectification of trinocular and multi-ocular stereovision system.

Chapter 4 Rectification of single-lens trinocular and multi-ocular stereovision system

In Chapter 3, we have introduced the purpose and the advantages that the rectification offers for single-lens binocular stereovision system. The associated algorithm, which is basically a geometry-based approach, has also been presented and verified experimentally. In this Chapter, we extend this approach to handle the rectification of single-lens trinocular and multi-ocular stereovision systems. In essence, in a two view system, we only consider the epipolar constraints between two images, but in a multi-view system, we need to consider the constraints amongst all views (refer to Appendix B). In addition, there would be more extrinsic parameters to be determined. This chapter is organized as follows. In section 4.1, we provide a detailed overview of our method, namely, the geometry-based approach for three-view image rectification, which includes virtual camera generation and computation of the intrinsic and extrinsic parameters of virtual cameras using geometrical analysis. The multi-ocular stereo vision rectification is presented in section 4.2. The experimental results and discussion are shown in section 4.3. The summary is given in the final section 4.4.

4.1 A geometry-based approach for three-view image rectification

Figure 4.1 shows the system setup of a single-lens trinocular (3F filter or tri-prism) stereovision system. Any image captured using this system is divided into three sub-images, which are called image triplet. These three sub-images are equivalent to three images captured using three virtual cameras which are created by the 3F filter. The image triplet is captured simultaneously (in one shot), and hence a dynamic scene can be handled by this system without any problem. However, we shall only consider static scene in our experiment.

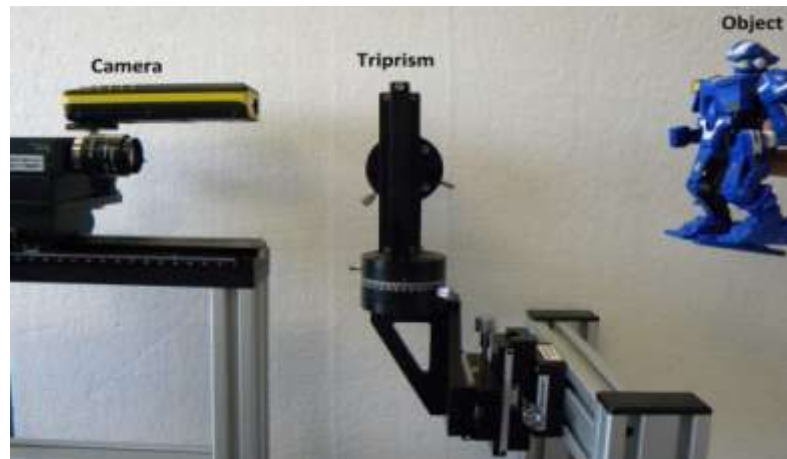


Figure 4.1 Single-lens based stereovision system using tri-prism

4.1.1 Generation of three virtual cameras

The key issue in modeling the single-lens trinocular stereovision system is to determine the extrinsic parameters of the virtual cameras. If a 3F filter is vertically positioned in front of a CCD camera as shown in Figure 4.2, in which the shape of the 3F filter is also illustrated, the image plane of this camera will capture three different views of the same scene behind the filter in one shot. These three sub-images are captured by three virtual cameras which are generated by the 3F filter. Two sample images captured by this system are shown in Figures 4.13 and 4.14 in Section 4.3. There are significant differences among the three sub-images as they are captured from different view angles and view scopes of the virtual cameras. Each virtual camera consists of one unique optical center and one “planar” image plane.

In our ensuing analysis, the following assumptions are made:

- 1) The real image plane of the CCD camera has consistent properties, such as pixel size, and focal length;
- 2) The 3F filter is exactly symmetrical with respect to the three apex edges and its center axis, which passes through the prism vertex and is normal to its back plane;

3) The back plane of the 3F filter is positioned such that it is parallel to the real camera image plane, and;

4) The projection of the 3F filter vertex on the camera image plane is located at the camera principle point (or the centre of the real camera image plane) and the projection of the three apex edges of the filter on the real camera image plane divides it into three sub-images equally.

With the aforesaid assumptions satisfied, the camera optical axis will pass through the 3F filter apex; the three virtual cameras will have identical properties and will be symmetrically located with respect to the real camera optical axis. Thus, the analysis of any one of the virtual camera would be sufficient as the results can be transposed to the other two virtual cameras. The three sub-regions of the image plane (and also the three corresponding virtual cameras) can be differentiated by using labels l , r and b which stand for left, right and bottom, respectively, as shown in Figure 4.2.

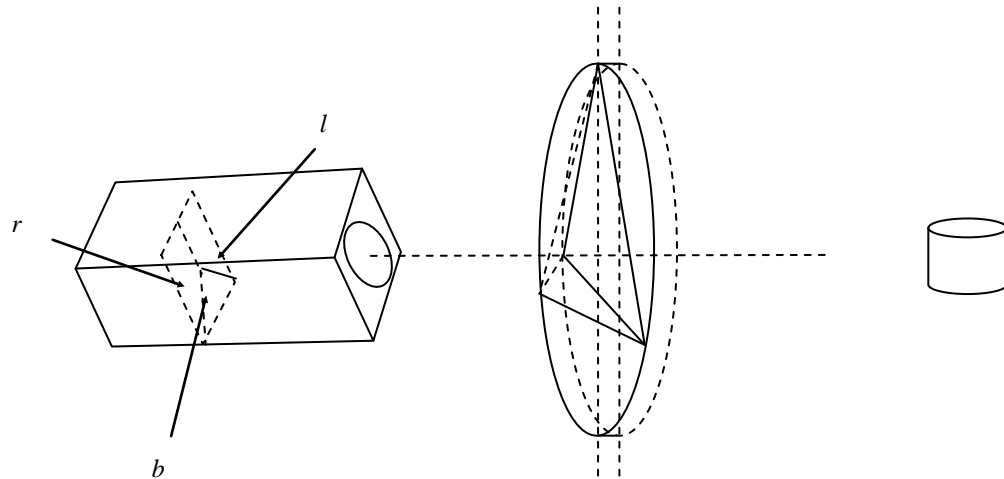


Figure 4.2 Single-lens stereovision system using 3F filter

4.1.2 Determination of the virtual cameras' projection matrix by geometrical analysis of ray sketching

The geometrical approach which is introduced in Chapter 3 is extended to study the virtual camera calibration of the 3F filter in this section. In our analysis and experiment, the camera lens distortions are assumed to be negligible.

1) The basic idea

For the setup shown in Figure 4.2, the sensor size and resolution of the camera CCD chip, geometry of the 3F filter, and also its relative position with respect to the real camera are known. The analysis of the 3F filter will be based on the schematic and ray diagram shown in Figure 4.3 and 4.4, respectively. Referring to the figure 4.4, given a point P on the right real camera image plane, the line joining point P and the focal point O intersects the triangle plane AO_2B of the 3F filter at point M , and the ray PM will be refracted twice through the 3F filter and will exit as ray NL (point N is on plane $A'A'_1$).

Next, consider the ray O_3O_2 , where point O_3 is the real camera image plane center and point O_2 is the 3F filter's vertex, this ray is refracted twice through the 3F filter and becomes ray JS (point J is on the plane $A'A'_1$). As the ray O_3O_2 coincides with the real camera optical axis, this indicates that the refracted ray JS corresponds to the left virtual camera optical axis as shown in Figure 4.4. By back-extending the ray NL and JS , their intersection, O_5 , can be found, which is the optical center (see Figure 4.4) of the left virtual camera. This intersection always exists as the rays NL and JS are coplanar. The coordinates of the optical center (O_5) will form the translational transformation matrix of the left virtual camera from with respect to the pre-defined world coordinate frame.

The implementation of this approach is not difficult. For example, to determine the equation of ray MN , the equation of line PM and the normal vector of plane AO_2B are computed first. Then we applied the parallelogram rule to obtain the direction vector of ray MN , knowing which we can compute the equation of ray MN . Other rays and points can be determined in the similar manner.

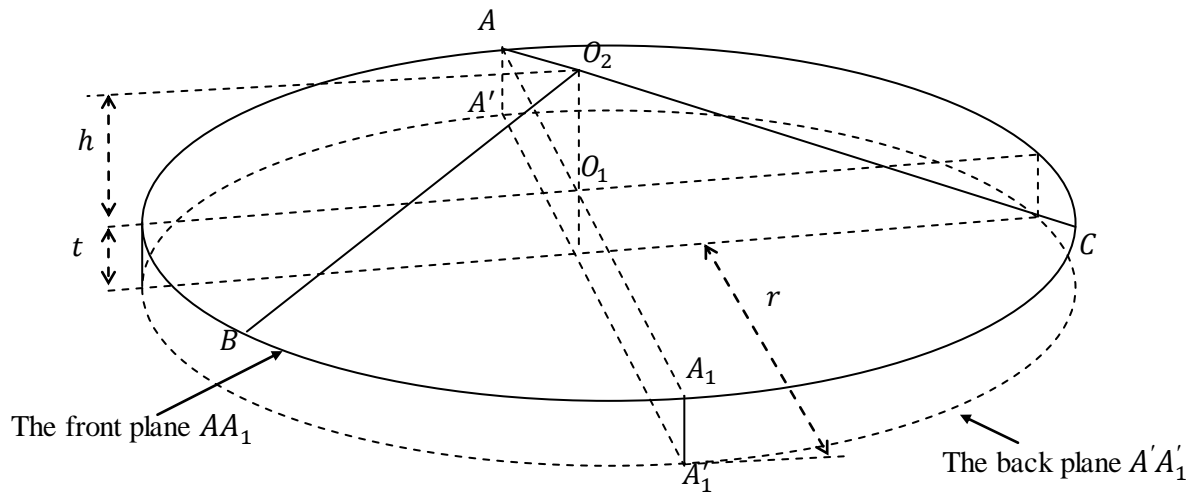


Figure 4.3 The structure of tri-prism

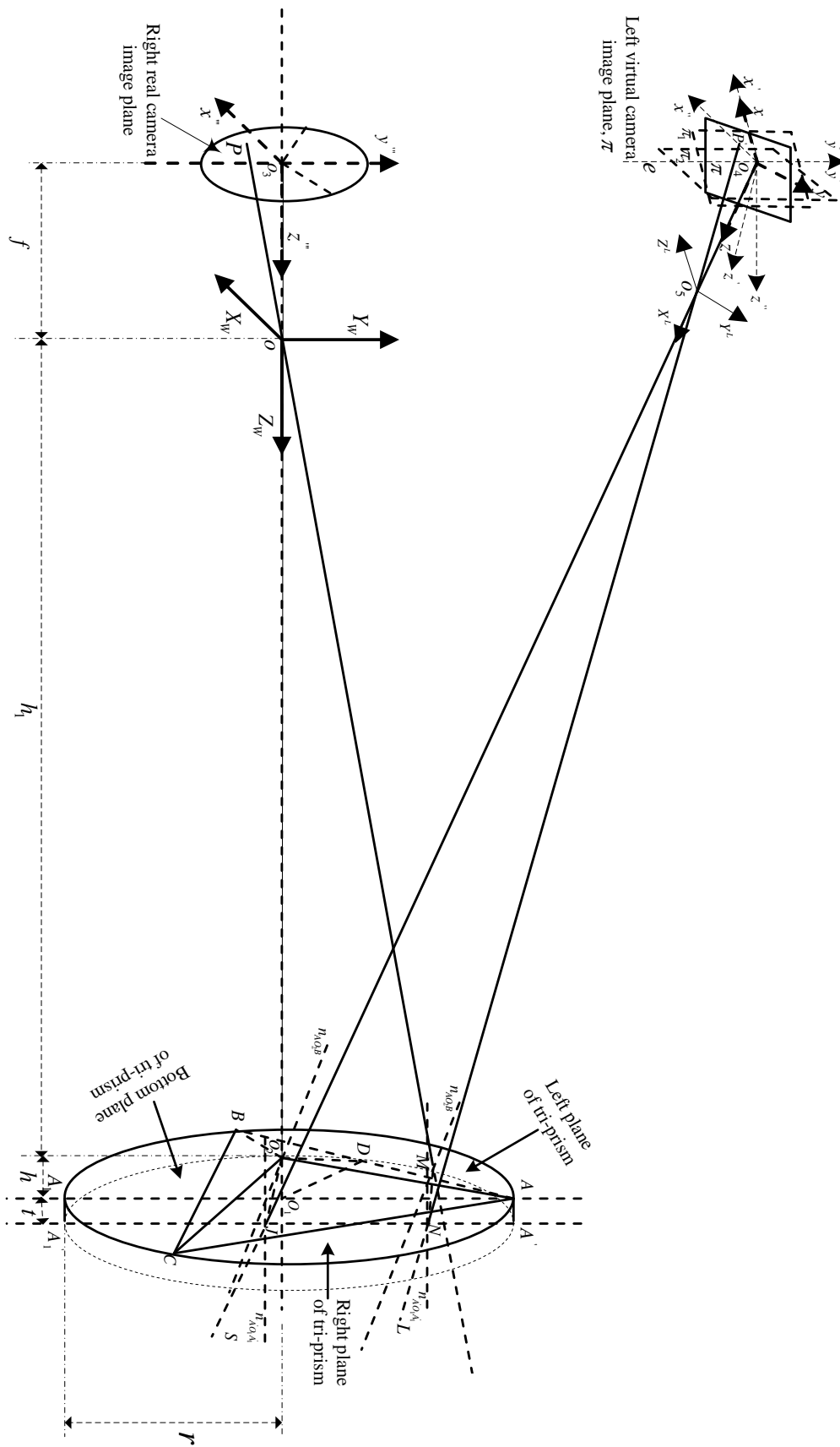


Figure 4.4 Geometry of left virtual camera using tri-prism

2) Compute the virtual cameras extrinsic parameters

In Chapter 3, we stated that the objective of camera calibration is to determine the camera intrinsic and extrinsic parameters. In this section, we shall compute mainly the extrinsic parameters while the intrinsic parameters can be obtained from the system hardware specification. The formation of virtual camera is illustrated in Figure 4.4.

In order to obtain the virtual camera extrinsic parameters, we follow two paths; path 1 to obtain the equation of line NL and path 2 to obtain the equation of line JS . The work flow is shown schematically in Figure 4.5. We then compute the two angles, γ and β , of the left and right virtual image plane and the angle, β , of the bottom virtual image plane with respect to real camera image plane (they are shown in Figures 4.4, 4.8 and 4.10, and will be explained fully later).

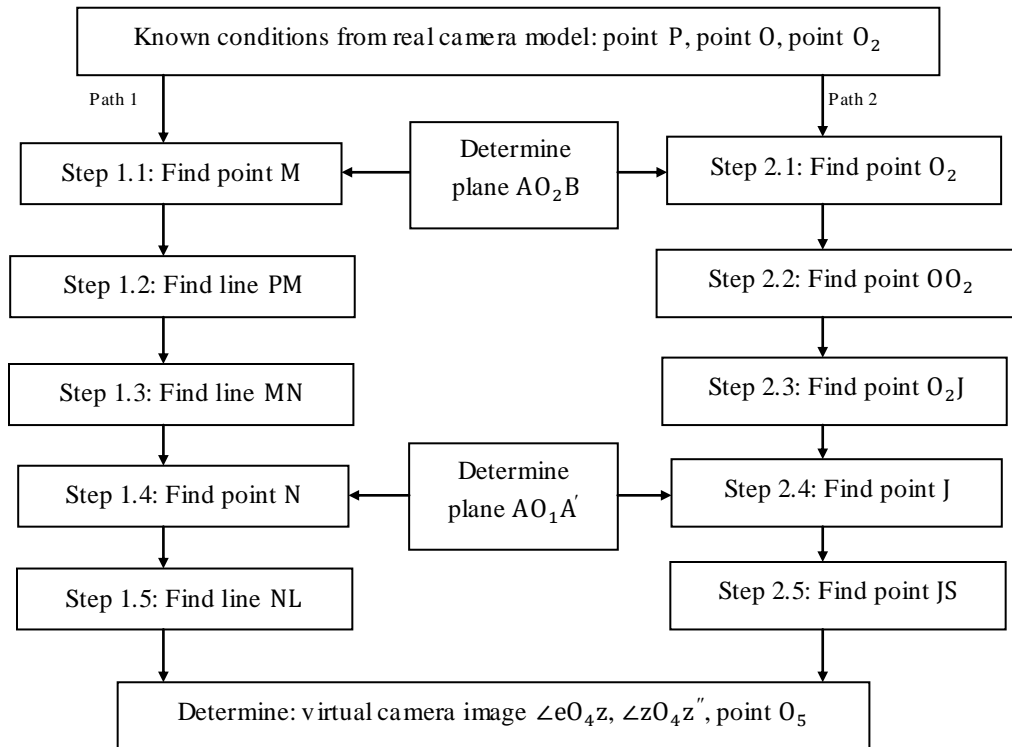


Figure 4.5 The workflow of determining the extrinsic parameters of virtual camera via geometrical analysis

Table 4.1 The parameters of tri-prism used in our setup

parameters	definition
h_1	The distance between the real camera optical center and the tri-prism apex (see Figure 4.3)
f	the focal length of the real cameras
f_v	the focal length of the virtual cameras
n	the refractive index of the tri-prism glass material
t	the thickness of tri-prism's front plane to back plane (see Figure 4.3)
h	The distance between 3F filter apex and its front plane of tri-prism (see Figure 4.3)
r	The radius of the tri-prism's back plane $A'A'_1$ (see Figure 4.3)
θ_1	The angle of incidence formed by line PM to the plane AO_2B (see Figure 4.4)
θ'_1	The angle of refraction formed by line PM passes the plane AO_2B (see Figure 4.4)
θ_2	The angle of refraction formed by line MN passes the plane $A'A'_1$ (see Figure 4.4)
θ'_2	The angle of incidence formed by line MN to the plane $A'A'_1$ (see Figure 4.4)

To determine the extrinsic parameters of the virtual cameras, the definition of the parameters are shown in Table 4.1 (refer to Figure 4.4):

Then, the steps to obtain the rotation matrix, translation vector and the optical centre of virtual camera through the two paths are shown (see Figure 4.5) in the following sections:

(1) Path 1: determine the equation of the prism plane

General plane equation in world coordinate frame can be expressed by the following equation:

$$T_x X + T_y Y + T_z Z = 1 \quad (4.1)$$

where $T = [T_x, T_y, T_z]$ is the normal vector of the plane.

Referring to Figure 4.4, the coordinates of the four vertices of the tri-prism with respect to the world coordinates are: $A(0, r, h_1 + h)$, $B(-\frac{\sqrt{3}}{2}r, -\frac{1}{2}r, h_1 + h)$, $C(\frac{\sqrt{3}}{2}r, -\frac{1}{2}r, h_1 + h)$, and $O_2(0, 0, h_1)$ the three equations of the three following planes can be determined as shown below:

Plane AO_2B :

$$\frac{\sqrt{3}h}{rh_1}X - \frac{h}{rh_1}Y + \frac{1}{h_1}Z = 1 \quad (4.2)$$

This process is repeated to obtain the equations of the planes AO_2C and BO_2C .

(2) Path1: determine the equation of the line PM

An arbitrarily point $P(x_P, y_P, z_P)$, is chosen on the right real camera image plane where $z_P = -f$; and $O(0, 0, 0)$ is the optical center of the real camera; therefore, the equation of line PM can be expressed as follows:

$$\frac{X}{-x_P} = \frac{Y}{-y_P} = \frac{Z}{f} \quad (4.3)$$

(3) Path1: determine the equation of the line MN

Point $M(x_m, y_m, z_m)$, which is at the intersection of the line PM and plane AO_2B , can be derived from the line and plane equation. Consider Eq. (4.2) and Eq. (4.3), the coordinates of M is given below:

$$\begin{cases} x_m = \frac{rh_1 x_P}{\sqrt{3}h x_P - h y_P - rf} \\ y_m = \frac{rh_1 y_P}{\sqrt{3}h x_P - h y_P - rf} \\ z_m = \frac{-rh_1 f}{\sqrt{3}h x_P - h y_P - rf} \end{cases} \quad (4.4)$$

Since plane AO_2B is known (Eq. (4.2)), its normal vector can be determined easily as follows:

$$N_{AO_2B} = \left[\frac{\sqrt{3}h}{rh_1}, -\frac{h}{rh_1}, \frac{1}{h_1} \right]$$

Thus, its unit vector is given by:

$$n_{AO_2B} = \frac{N_{AO_2B}}{|N_{AO_2B}|} = \left[\frac{\sqrt{3}h}{\sqrt{4h^2 + r^2}}, -\frac{h}{\sqrt{4h^2 + r^2}}, \frac{r}{\sqrt{4h^2 + r^2}} \right] \quad (4.5)$$

The direction vector of the line $\overrightarrow{PM} = [-x_p, -y_p, f]$; and its unit vector is

$$n_{\overrightarrow{PM}} = \frac{\overrightarrow{PM}}{|\overrightarrow{PM}|} = \left[\frac{-x_p}{\sqrt{x_p^2 + y_p^2 + f^2}}, \frac{-y_p}{\sqrt{x_p^2 + y_p^2 + f^2}}, \frac{f}{\sqrt{x_p^2 + y_p^2 + f^2}} \right] \quad (4.6)$$

From Eq. (4.5) and (4.6), we can obtain the angle θ_1 which is an incidence angle of line PM formed with the normal N_{AO_2B} on the plane AO_2B of the tri-prism

$$n_{AO_2B} \cdot n_{\overrightarrow{PM}} = |n_{AO_2B}| |n_{\overrightarrow{PM}}| \cos \theta_1 = \cos \theta_1 \quad (4.7)$$

From Eq. (4.7), we can then obtain the value of θ_1 :

$$\theta_1 = \arccos \left(\frac{-\sqrt{3}h x_p + h y_p + r f}{\sqrt{4h^2 + r^2} \sqrt{x_p^2 + y_p^2 + f^2}} \right) \quad (4.8)$$

θ'_1 is the refracted angle of the line PM at plane AO_2B . According to the law of refraction:

$$\frac{\sin \theta_1}{\sin \theta'_1} = n$$

θ'_1 is thus given as

$$\theta'_1 = \arcsin \left(\frac{\sin \theta_1}{n} \right) \quad (4.9)$$

The unit direction vector $n_{\overrightarrow{MN}} = [x_{\overrightarrow{MN}}, y_{\overrightarrow{MN}}, z_{\overrightarrow{MN}}]$, where N is the intersection of line MN and the back plane of the tri-prism $Z = h_1 + h + t$.

To find the equation of the line MN , the parallelogram rule is employed as shown in Figure 4.6. The figure shows the relationship between the unit direction vectors n_{AO_2B} and $n_{\overline{PM}}$.

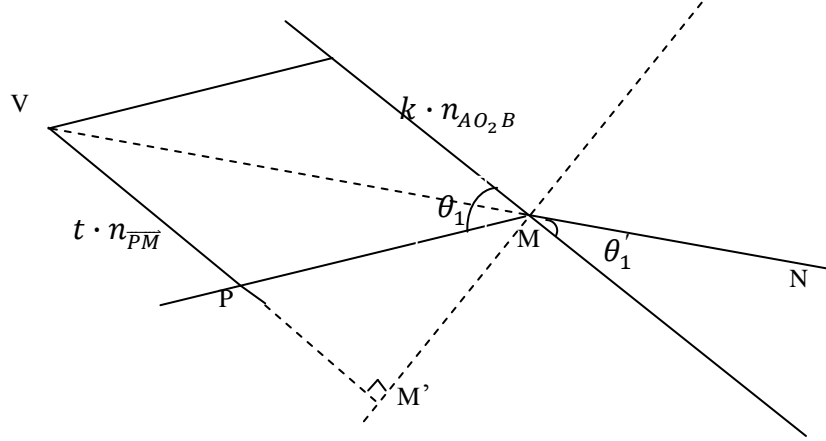


Figure 4.6 Relationship of direction vector line PM
and direction normal vector of plane AO_2B

Referring to Figure 4.6 and considering the parallelogram rule, the unit direction vector of MN is expressed by both the unit direction vector of n_{AO_2B} and $n_{\overline{PM}}$,

$$n_{\overline{MN}} = k \cdot n_{AO_2B} + t \cdot n_{\overline{PM}} \quad (4.10)$$

From Figure 4.6, we can obtain the equations as

$$\begin{cases} |VM'| = |n_{\overline{VM}}| \cdot \cos \theta'_1 \\ |VM'| = |VP| + |PM'| = |k \cdot n_{AO_2B}| + |PM'| \\ |PM'| = |m \cdot n_{AO_2B}| \\ |MM'| = |n_{\overline{VM}}| \cdot \sin \theta'_1 = |t \cdot n_{\overline{PM}}| \cdot \sin \theta_1 \\ |VM| = |n_{\overline{MN}}| \end{cases} \quad (4.11)$$

where k, t , and m are scalars.

From Eq. (4.11), the solution of k, t , and m are expressed as follows:

$$\begin{cases} k = \frac{\sin\theta'_1}{\sin\theta_1} \\ m = \frac{\sin\theta'_1 \cdot \cos\theta_1}{\sin\theta_1} \\ t = \frac{\sin(\theta_1 - \theta'_1)}{\sin\theta_1} \end{cases} \quad (4.12)$$

Thus, the unit direction vector of MN is described by

$$n_{\overline{MN}} = \frac{\sin\theta'_1}{\sin\theta_1} \cdot n_{AO_2B} + \frac{\sin(\theta_1 - \theta'_1)}{\sin\theta_1} \cdot n_{\overline{PM}} \quad (4.13)$$

Hence, the line MN equation is given as:

$$\frac{X - x_m}{x_{\overline{MN}}} = \frac{Y - y_m}{y_{\overline{MN}}} = \frac{Z - z_m}{z_{\overline{MN}}} \quad (4.14)$$

After having obtained the equation of line MN , the point $N = [x_N, y_N, z_N]$ is

$$\begin{cases} x_N = \frac{((h_1 + h + t) - z_m) \cdot x_{\overline{MN}}}{z_{\overline{MN}}} + x_m \\ y_N = \frac{((h_1 + h + t) - z_m) \cdot y_{\overline{MN}}}{z_{\overline{MN}}} + y_m \\ z_N = h_1 + h + t \end{cases} \quad (4.15)$$

(4) Path1: determine the equation of the line NL

Since the equation of the line MN is known, we can compute the equation of the line NL . We first compute the angle θ'_2 , which is formed by the line MN and the normal vector to the plane AO_2B , N_{AO_2B} . The angle of refraction of the line MN at the plane $A_1A'_1$, θ_2 , can be obtained in Eq. (4.17). $n_{\overline{Z_w}}$ is the unit direction vector of the Z_w -axis.

$$n_{\overline{MN}} \cdot n_{\overline{Z_w}} = \cos(\theta'_2)$$

$$\frac{\sin\theta_2}{\sin\theta'_2} = n \quad (4.16)$$

$$\theta_2 = \arcsin(n \cdot \sin\theta'_2) \quad (4.17)$$

The direction vector of \overline{NL} is obtained using the similar procedure as in the case of the direction vector of \overline{MN} .

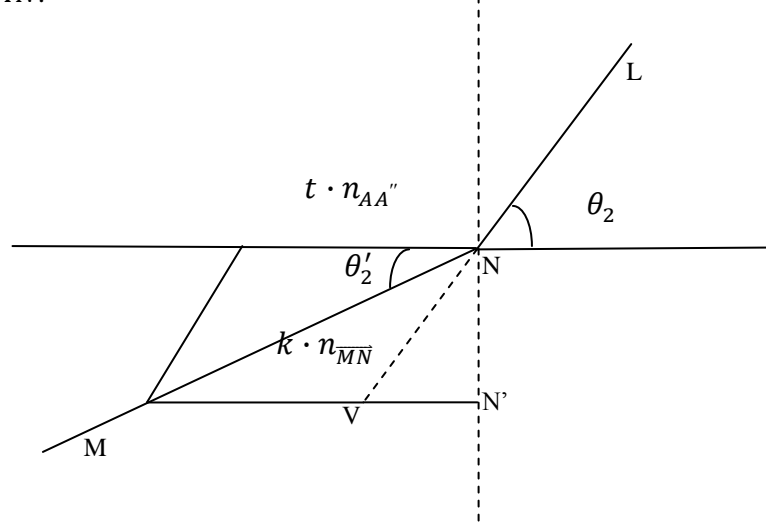


Figure 4.7 Illustration of direction vector of line MN and the direction normal vector of plane $A_1A'_1$

We write:

$$n_{\overline{NL}} = k \cdot n_{\overline{MN}} - t \cdot n_{AA''} \quad (4.18)$$

where k , and t are scalars.

By referring to Figure 4.7, we obtain the equations

$$\begin{cases} |NN'| = |k \cdot n_{\overline{MN}}| \cdot \sin\theta'_2 \\ |MN'| = |MV| + |VN'| = |t \cdot n_{AA''}| + |n_{\overline{VN}}| \cdot \cos\theta_2 \\ |NN'| = |n_{\overline{VN}}| \cdot \sin\theta_2 \\ |MN'| = |k \cdot n_{\overline{MN}}| \cdot \cos\theta'_2 \\ |VN| = |n_{\overline{NL}}| \end{cases} \quad (4.19)$$

From Eq.(4.19), the solution of k and t are expressed

$$\begin{cases} k = \frac{\sin\theta_2}{\sin\theta'_2} \\ t = \frac{\sin(\theta_2 - \theta'_2)}{\sin\theta'_2} \end{cases} \quad (4.20)$$

Thus, the unit direction vector of MN is given

$$n_{\overline{NL}} = \frac{\sin\theta_2}{\sin\theta'_2} \cdot n_{\overline{MN}} - \frac{\sin(\theta_2 - \theta'_2)}{\sin\theta'_2} \cdot n_{\overline{Z_w}} \quad (4.21)$$

Finally, we obtain the equation of line NL

$$\frac{X - x_N}{x_{\overline{NL}}} = \frac{Y - y_N}{y_{\overline{NL}}} = \frac{Z - z_N}{z_{\overline{NL}}} \quad (4.22)$$

(5) Path2: determine the equation of the line JS

After having followed path 1, we now determine the equation of line JS (Figure 4.4) along path 2. The steps involved are very similar with those of path 1. Thus, we shall repeat the same procedure of path 1 with another different image point $O_3(0,0,-f)$. The detailed steps would not be repeated here.

Finally, the equation of line JS is

$$\frac{X - x_J}{x_{\overline{JS}}} = \frac{Y - y_J}{y_{\overline{JS}}} = \frac{Z - z_J}{z_{\overline{JS}}} \quad (4.23)$$

where the point $J = [x_J, y_J, z_J]$ is on the back plane of tri-prism and $n_{\overline{JS}} = [x_{\overline{JS}}, y_{\overline{JS}}, z_{\overline{JS}}]$ is the unit direction vector of JS .

(6) Determine the optical centre of the virtual camera O_5

After following the two paths mentioned above, and having obtained the equation of line NL and JS , we can now determine the optical center of the left virtual camera, which is the intersection of these two lines.

Let $O_5(x_{O_5}, y_{O_5}, z_{O_5})$ denote the optical center of the virtual camera which can be obtained from the intersection of these two lines:

$$\begin{cases} \frac{X - x_N}{x_{NL}} = \frac{Y - y_N}{y_{NL}} = \frac{Z - z_N}{z_{NL}} \\ \frac{X - x_J}{x_{JS}} = \frac{Y - y_J}{y_{JS}} = \frac{Z - z_J}{z_{JS}} \end{cases} \quad (4.24)$$

Theoretically the lines JS and NL should always intersect, however due to digitization errors, this may not necessary be the case. Therefore, an approximation of the coordinates of O_5 is necessary. The computation the point O_5 using mid-point theorem is presented in the Appendix A.

(7) Determine the rotation angles α and β

The computation of the rotation angles of the virtual camera image plane with respect to the real camera image plane is presented as follows (see Figure 4.4):

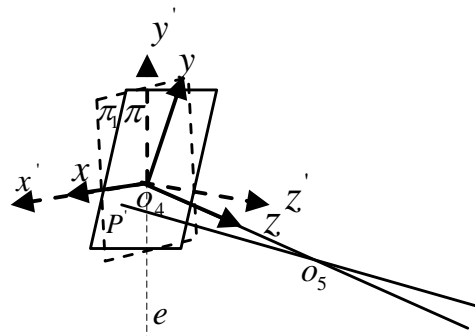


Figure 4.8 The virtual image plane π rotated to image plane π_1 about x -axis

Refer to Figure 4.4 and Figure 4.8, the angle $\angle eO_4z$ can be computed by the following steps:

$$n_{\overline{JS}} \cdot n_{\overline{Yw}} = |n_{\overline{JS}}| |n_{\overline{Yw}}| \cos \varphi = \cos \varphi \quad (4.25)$$

where $\varphi = \angle eO_4z$ and $n_{\overline{JS}}$ is the unit direction vector of JS .

Thus,

$$\gamma = \frac{\pi}{2} - \varphi = \frac{\pi}{2} - \arccos(n_{\overline{JS}} \cdot n_{\overline{Yw}}) \quad (4.26)$$

where γ is the angle of rotation of the left virtual image plane π to the new image plane π_1 about x -axis.

Then, we obtain the new axis, z' -axis which is obtained by rotating the z -axis at an angle of γ about the x -axis (Figure 4.8 and 4.9), and we also utilize the parallelogram rule to obtain unit vector of z' -axis, $n_{\overline{z'}}$,

$$n_{\overline{z'}} = \frac{1}{\cos(\gamma)} n_{\overline{JS}} - \frac{1}{\cot(\gamma)} n_{\overline{Yw}} \quad (4.27)$$

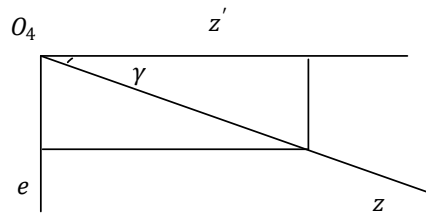


Figure 4.9 The relationship of z' -axis and z -axis

Next, by referring to Figure 4.10, the angle of rotation of the image plane π_1 to image plane π_2 is computed

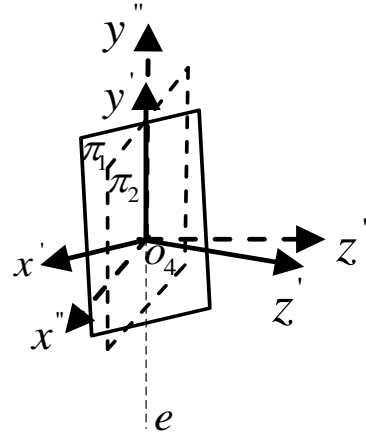


Figure 4.10 The image plane π_1 rotates to image plane π_2 about y' -axis

$$n_{z'} \cdot n_{z''} = \cos\beta \quad (4.28)$$

where $\beta = \angle z'O_4z''$ is the angle of rotation of π_1 to the new image plane π_2 about y' -axis.

(8) Determine R_L and T_L

After determining the orientation of the left virtual camera's image plane and its optical centre, we shall obtain the rotational matrix R_L which is the rotational transformation of an image point on the real image plane to the left virtual camera image plane, and translation vector T_L which is the translational transformation of an image point on the real image plane to the left virtual camera image plane.

$$\text{Rotation matrix, } R_L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \\ 0 & -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$\text{Translation vector, } T_L = \begin{bmatrix} x_{O_5} \\ y_{O_5} \\ z_{O_5} \end{bmatrix}$$

The camera intrinsic matrix M of our stereovision system is known.

$$\text{Camera intrinsic matrix, } M = \begin{bmatrix} -f_x & 0 & C_x \\ 0 & -f_y & C_y \\ 0 & 0 & 1 \end{bmatrix}$$

where:

f_x , focal length of the virtual camera in the x -direction in terms of pixel size

f_y , focal length of the virtual camera in the y -direction in terms of pixel size

C_x , the coordinates of the image point at the centre of the virtual camera image plane in the x -direction C_y , the coordinates of the image point at the centre of the virtual camera image plane in the y -direction

The same procedure is used to obtain the rotational matrix and translation vector of the right and bottom virtual camera image plane which are the rotational and translational transformation of image points on the real image plane to the virtual camera image planes. However, based on the geometry of the bottom plane of the tri-prism with respect to the real camera image plane, we find that there is only one rotation angle with respect to the real camera image plane (see Figure 4.4). Thus the rotational matrix is

$$R_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \\ 0 & -\sin\gamma & \cos\gamma \end{bmatrix} \quad (4.29)$$

The projection matrixes which relate the 3D coordinates to 2D coordinate of the virtual camera image plane is given by [10]:

$$P_L = M[R_L|T_L], \quad P_R = M[R_R|T_R], \quad P_B = M[R_B|T_B] \quad (4.30)$$

Finally, we can obtain the perspective projection matrix P_L , P_R , and P_B which are the left, right, and bottom virtual camera, respectively. R_R and R_B are the rotation matrix of the right and bottom virtual camera image planes which are the rotational transformation of image points on the real image plane to the right and bottom virtual camera image planes. T_R and T_B are the translation vector of the right and bottom virtual camera image planes which are the translational transformation of image points on the real image plane to the right and bottom virtual camera image planes.

4.1.3 Rectification Algorithm

We have introduced rectification algorithm for each of the virtual cameras in Section 3.2.2.

The new virtual camera coordinates is defined as

$$R_{rect} = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix} \quad (4.31)$$

r_1 is the new X -axis of the virtual camera which is chosen to be parallel to X_w -axis,

r_2 is the new Y -axis of the virtual camera which is same as the old Y_w -axis,

r_3 is the new Z -axis of the virtual camera which is the cross product of r_1 and r_2 .

The final rectification transformation matrix:

$$R_l = R_L R_{rect}, \quad R_r = R_R R_{rect}, \quad R_b = R_B R_{rect} \quad (4.32)$$

The steps of the rectification algorithm in this case are the same to the one explained and defined in Section 3.2.2.

4.2 The multi-ocular stereo vision rectification

The terms *multi-ocular* and *multi-view* often have the same implications which appear frequently in recent literatures [96, 97, 98]. These two terms are normally used to describe a series of images captured from the same scene. Multi-ocular or multi-view images normally provide more comprehensive information on the environment and have attracted a great amount of interest.

A *multi-camera* system is generally required when the multi-view or multi-ocular images are required to be captured simultaneously. However, the camera setup, calibration and synchronization of a multi-camera system are usually more difficult and complicated than typical single camera or two camera vision system. Some discussions on multi-camera calibration are reported in [96, 97, 98].

This section presents the analysis and discussion on the rectification issue of a single-lens multi-ocular stereovision system. This system is able to capture more than three different views of the same scene simultaneously using only one real camera with the aid of a multi-face prism. It combines the advantages of a single-lens stereovision and multi-ocular stereovision system. Dynamic scene image capturing or video rate image capturing is not a problem for this system.

Each image captured by this single-lens system can be divided into more sub-images and these sub-images can be taken as images taken by multiple virtual cameras which are created by the multi-face prism. Two approaches are used to analyze this multi-ocular system: the first one is based on the calibration technique and the second one is based on geometrical analysis of ray sketching. The geometrical based approach attracts greater interest because of its simpler implementation. It does not require the usual complicated calibration process, and several points of the real image plane are enough to determine the virtual cameras system

once the system setup is fixed; in addition, simple pin-hole camera model is used to analyze the system [21].

We have analyzed the single-lens stereovision system using bi-prism and tri-prism. The rectification of the stereovision system can be handled adequately and efficiently by our geometry-based approach. Here, we will extend the approach in the analysis of the single-lens multi-faced prism base stereovision system.

Figure 4.11 shows the geometry of the single-lens based stereovision system using 4-face prism. We use the same steps used in our geometry-based approach (section 4.1) to determine the rotation matrix R and translation vector T of virtual camera image planes which are the rotational and translational transformation of image points on the real image plane to the virtual camera image planes.

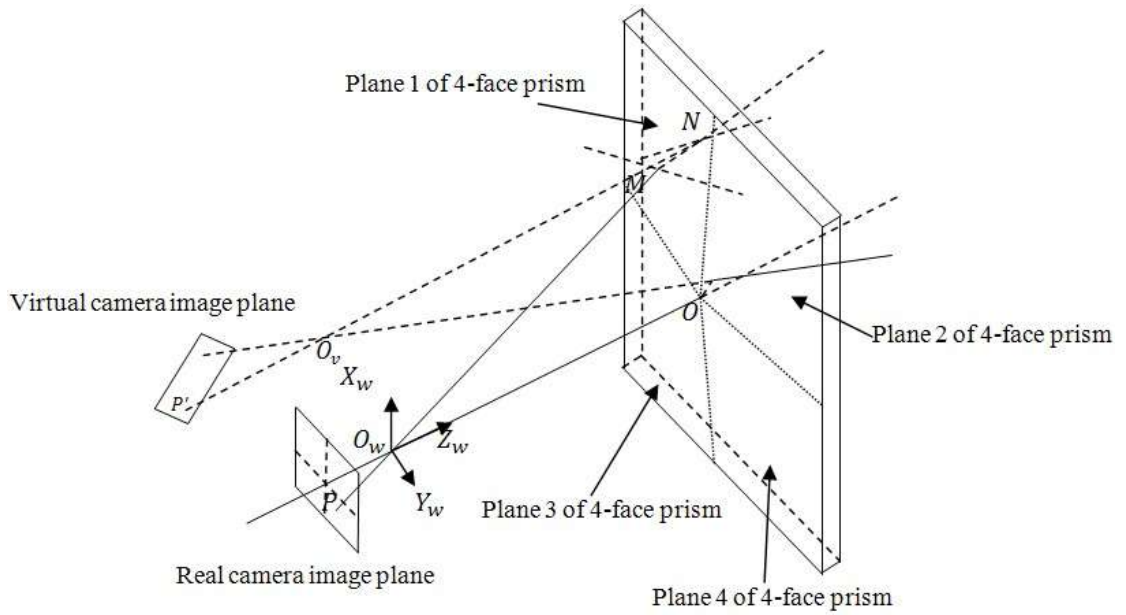


Figure 4.11 Geometry of single-lens based on stereovision system using 4-face prism

By analyzing the geometry of the single-lens stereovision system using 4-face prism, we can determine the angles of one of the virtual camera image planes (take plane 1 of 4-face prism

as example, Figure 4.11) with respect to the real camera image plane using the same procedure as in the case of tri-prism based single-lens stereovision system. The rotational matrix R_1 and translation vector T_1 of the virtual camera image plane of plane 1 of 4-face prism, which are the rotational and translational transformation of image points on the real image plane to the virtual camera image plane are obtained as follows

$$\text{Rotational matrix, } R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma_1 & \sin\gamma_1 \\ 0 & -\sin\gamma_1 & \cos\gamma_1 \end{bmatrix} \begin{bmatrix} \cos\beta_1 & 0 & \sin\beta_1 \\ 0 & 1 & 0 \\ -\sin\beta_1 & 0 & \cos\beta_1 \end{bmatrix} \quad (4.33)$$

where γ_1 and β_1 are the rotation angles of the virtual camera image plane with respect to the selected axis (see Figure 4.8 and 4.10). The two angles can be determined using the steps illustrated in Figure 4.5.

$$\text{Translation vector, } T_1 = \begin{bmatrix} x_{O_v} \\ y_{O_v} \\ z_{O_v} \end{bmatrix} \quad (4.34)$$

where $O_v(x_{O_v}, y_{O_v}, z_{O_v})$ is one of the optical centre of the virtual camera (see Figure 4.11).

The computing method of O_v is the same as that of O_5 in Section 4.1.

The rotational matrix and translation vector of the virtual camera image plane of the other three planes of 4-face prism can be computed using the same procedure of the plane 1 of 4-face prism.

Then, we establish the rectification coordinate system of the new virtual cameras and rectification transformation matrix which move the epipole to infinity as described in section 4.1.3,

$$R_{rec} = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix} \quad (4.35)$$

r_1 is the new X -axis of the virtual camera which is chosen to be parallel to X_w -axis

r_2 is the new Y -axis of the virtual camera which is same as the old Y_w -axis

r_3 is the new Z -axis of the virtual camera which is the cross product of r_1 and r_2

The final rectification transformation matrix:

$$R_{L_up} = R_1 R_{rec}, \quad R_{L_down} = R_3 R_{rec}, \quad R_{R_up} = R_2 R_{rec}, \quad R_{R_down} = R_4 R_{rec} \quad (4.36)$$

The detailed steps of rectification are given in section 3.2.2.

From the analysis of the single-lens stereovision using bi-prism, tri-prism and 4-face prism, we can conclude that the geometry-based approach of rectification can be extended to multi-face prism (Figure 4.12 shows a system with a 5-face prism), and the same procedures of our geometrical method can be used to determine the rotational matrix and translation vector of virtual camera image planes which are the rotational and translational transformation of image points on the real image plane to the virtual camera image planes (refer to Section 4.1.2). Then, the new virtual camera coordinates is defined (refer to Section 4.1.3). Finally, the final rectification transformation matrices of multi-view virtual camera image planes are obtained and the detailed steps of rectification refer to Section 3.2.2.

Essentially, the algorithm of rectification using single-lens prism based stereovision system involves the following steps:

- (1) Utilize several points in the real camera image plane to determine the orientation of the virtual cameras (Section 4.1.1);
- (2) Determine the extrinsic parameters of the virtual camera using the geometry-based approach (Section 4.1.2);

- (3) Define the new virtual cameras coordinates (Section 4.1.3);
- (4) Formulate the rectification transformation matrix (Section 4.1.3);
- (5) Compute the projection point on the new virtual camera image planes which are rectified from the original virtual camera image planes (Section 3.2.2).

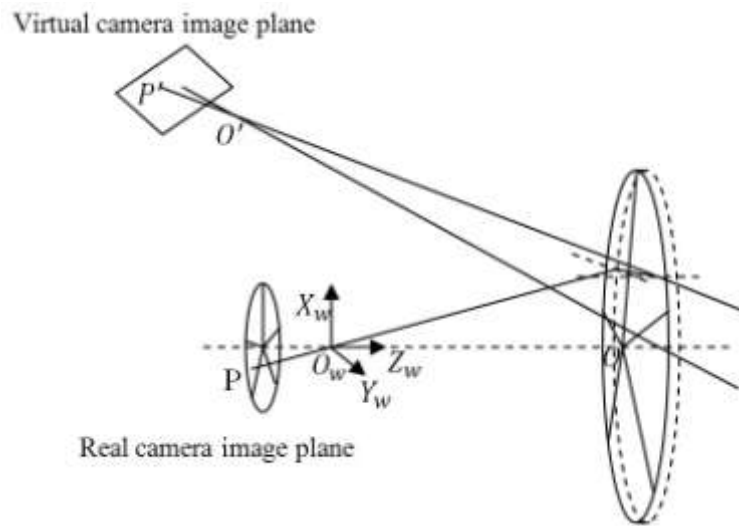


Figure 4.12 Geometry of the single-lens stereovision system using 5-face prism

4.3 Experimental results and discussion

(1) Experiments of single-lens stereovision system with a trip Prism

The system setup for conducting the experiments has been shown in Figure 4.1. We take two sets of images with different values of h (refer to Figure 4.3) of the tri-prism. The captured images of a robot and a red rose are shown in Figures 4.13(a) and 4.14 (a), respectively. We can see that each captured image has three sub-images in it, according to the geometry of the tri-prism used. In our case, they can be appropriately labeled as “left”, “right” and “bottom” images.

The parameters for the tri-prism used in the experiment are: $n = 1.5, r = 37.3, d = 210mm, h = 4mm, t = 6.8mm, f = 25mm, f_x = 25mm, f_y = 25mm, C_x = 384$ and $C_y = 521$.

Three points are selected in the left image and right image as shown in Figure 4.13 (b) and (d), respectively. The correspondence epipolar lines are then determined and plotted. They are: The epipolar lines based on the left and right image pairs are plotted in the right image (Figure 4.13 (d)). There are two sets of epipolar lines plotted in the bottom image (Figure 4.13(c)), one set is based on the left and bottom image pairs, while the other set is based on the right and bottom image pairs.

After the implementation of the rectification algorithm, we observe the followings

- the three selected points in the left and right image are rectified; and the epipolar lines are rectified (Figure 4.13 (e) and (g));
- the epipolar lines at the bottom image shown earlier in Figure 4.13 (c) are also rectified as shown in Figure 4.13 (f).

We can see that all the rectified epipolar lines become horizontal and along the scan-lines of the image. This observation verifies the correctness of our algorithm.

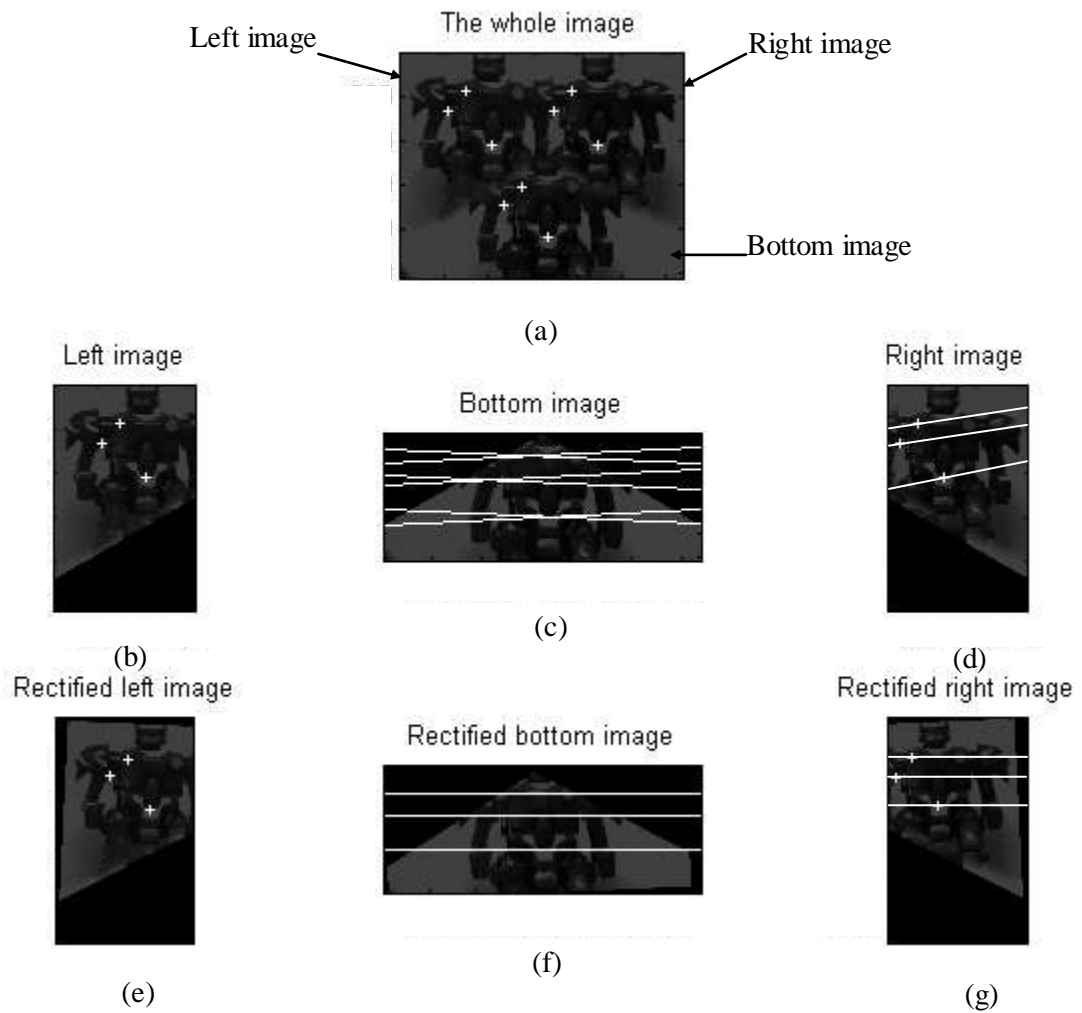


Figure 4.13 The image captured from trinocular stereovision and rectified images (robot)

We have also conducted another experiment using the captured images of a red rose using the same setup but with different prism height ($h = 6.2mm$). , Figure 4.14 shows the similar results, and the same observations can also be made. Therefore, our approach in rectification is verified.

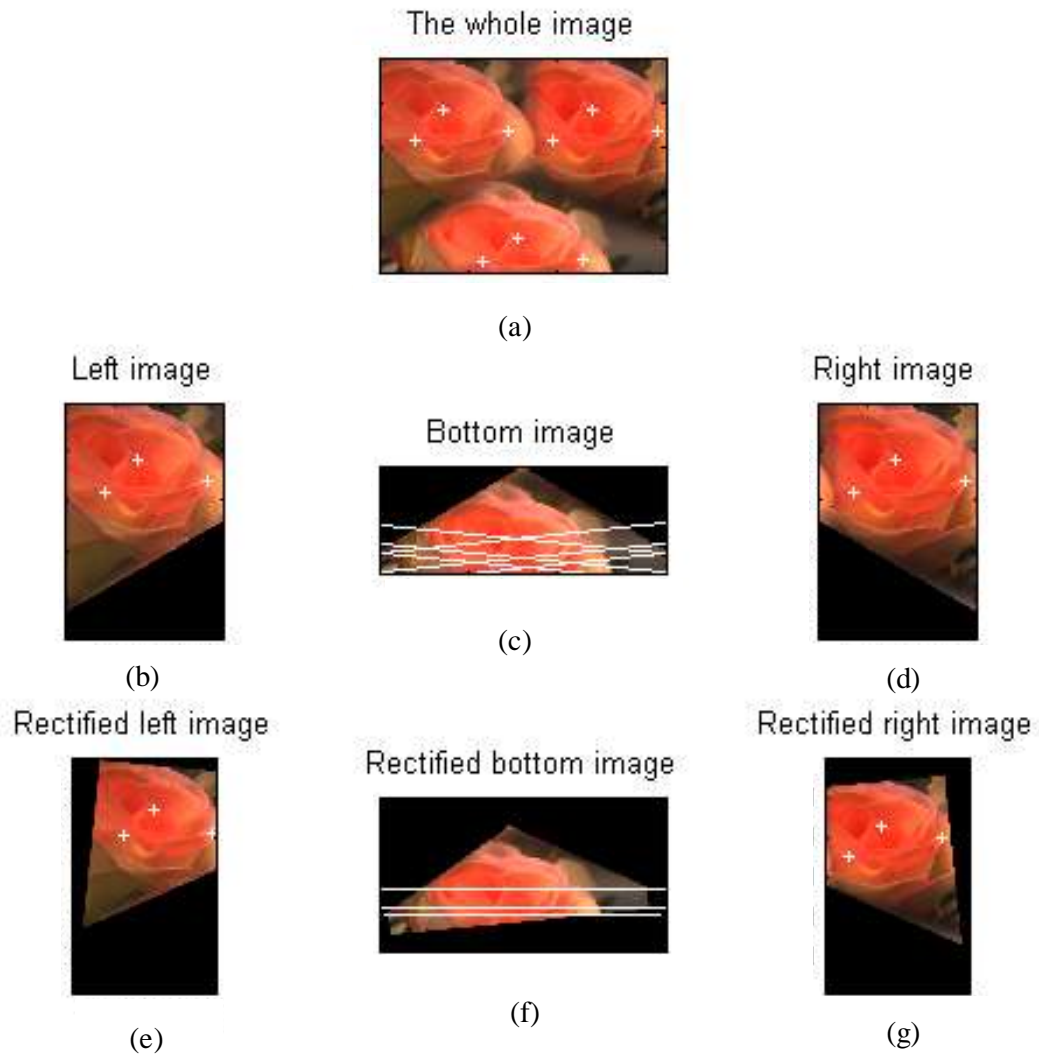


Figure 4.14 The image captured from trinocular stereovision and rectified images (red rose)

Table 4.3 shows the results of stereo correspondence using the conventional calibration method (Zhang's method) [29] and our geometrical approach. The following Table 4.2 gives the descriptions of the columns in Table 4.3. Note that the Euclidean distances calculated in column 6 and 7 indicate the errors in the determination of the correspondence points using the Geometrical approach and the conventional calibration method, respectively. For the 3rd column, the coordinates of correspondence points are the intersection of the epipolar lines of the left and right image point in the bottom image are obtained using geometrical approach;

while those in the 5th column the coordinates are of the correspondence points are obtained in the similar in the similar way except that conventional calibration method was used instead.

4.2 The descriptions of the columns in Table 4.3

Column number	Description
1	Label of test point
2	Coordinates of the points in the left and right images
3	Correspondence point in the bottom image obtained using geometrical approach
4	Coordinates of the correspondence point obtained by direct measurement
5	Correspondence point in the bottom image obtained using conventional calibration method.
6	Euclidean distance between the correspondence points determined by geometrical approach (Column 3) and direct measurement (Column 4)
7	Euclidean distance between the correspondence points determined by conventional calibration method (Column 5) and direct measurement (Column 4)

Table 4.3 shows that the values in column 6 are consistently lower than those in column 7. The average distances of correspondence point obtained by Geometrical approach and the conventional calibration method are 7.5036 and 12.8809, respectively. The average distance shows that the geometrical approach calibration is better than the conventional calibration method. This improvement can be attributed to the fact that the geometrical approach can obtain more accurate virtual camera extrinsic parameters than the conventional calibration method. This also shows that our proposed geometrical approach is comparatively better than the conventional calibration method in rectifying the virtual cameras. Furthermore, the results verify our rectification algorithm is indeed valid even for single-lens multi-view stereovision system.

Table 4.3 The result of comparing calibration method and geometry method for obtaining stereo correspondence

Test data label	Pixel coordinate (in the order of left, and right)	correspondence point coordinate by Geometrical Approach	Actual correspondence coordinate	Correspondence point coordinate by Calibration Approach	Distance (pixel, Geometry)	Distance (pixel, Calibration)
1	L(100, 150)	(425, 148)	(422, 150)	(413, 155)	3.6055	10.247
	R(740, 150)					
2	L(150, 210)	(472, 208)	(474, 213)	(468, 203)	5.3851	11.6619
	R(790, 210)					
3	L(216, 265)	(537, 253)	(544, 257)	(542, 248)	8.0623	9.2195
	R(856, 265)					
4	L(250, 350)	(571, 353)	(565, 355)	(578, 357)	6.3246	13.1529
	R(890, 350)					
5	L(300, 180)	(618, 177)	(615, 183)	(606, 174)	6.7082	12.7279
	R(940, 180)					
6	L(356, 384)	(681, 388)	(670, 387)	(684, 394)	11.0453	15.6525
	R(996, 384)					
7	L(416, 418)	(731, 423)	(736, 414)	(743, 427)	10.2956	14.7648
	R(1056, 418)					
8	L(510, 501)	(827, 498)	(834, 493)	(824, 505)	8.6023	15.6205
	R(1150, 501)					
AVG					7.5036	12.8809

(2) Experiments of single-lens stereovision system with a 4-face prism

Figure 4.11 shows our system setup with a 4-face prism. Figure 4.15 shows the original images and Figure 4.16 shows the four rectified images. Four points are selected and marked in Figure 4.15(a), and their epipolar lines are plotted in Figures 4.15(b), (c) and (d). After the implementation of the rectification algorithm, the four points are rectified in Figure 4.16(a), and their epipolar lines are shown in Figures 4.16(b), (c) and (d). We can again see that all the epipolar lines have been rectified. They are all along the horizontal scan lines of the images.

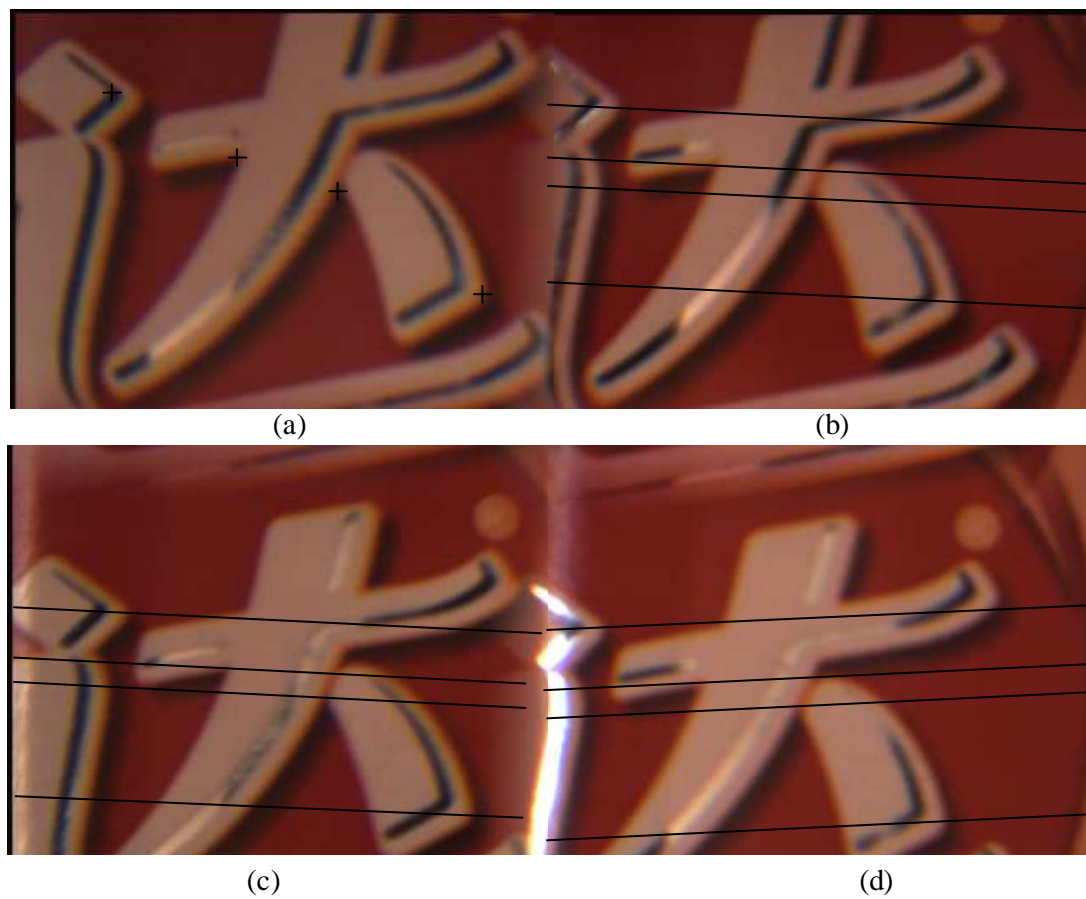


Figure 4.15 The images capture from four-ocular stereovision (“da” images)

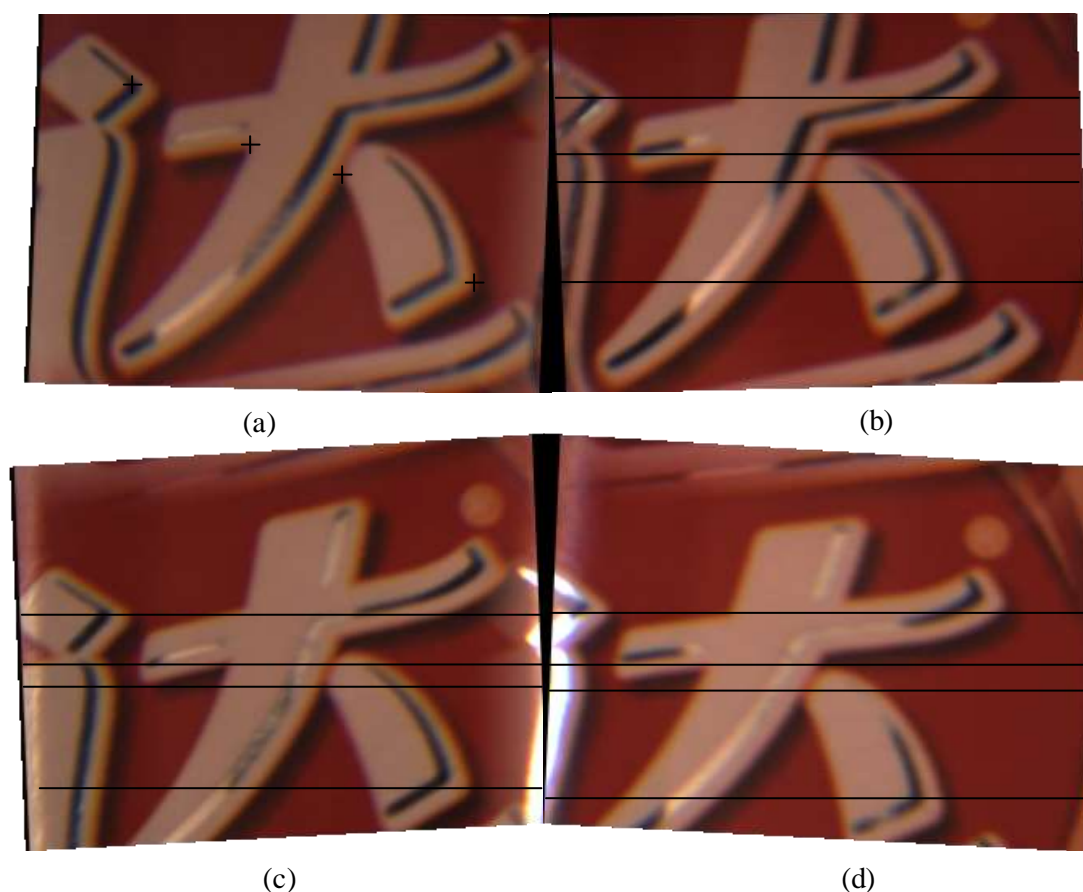


Figure 4.16 The images capture from four-ocular stereovision and rectified images (“da” images)

4.4 Summary

This chapter presents the algorithm of rectification using single-lens prism (tri-prism and multi-face prism) based stereovision system. The geometry-based approach which is proposed in Chapter 3 to solve the rectification issue is further extended to solve the similar problem on tri-prism/multi-prism based single-lens stereovision system. Next, the rectification issue of single-lens stereovision system using multi-face prism were briefly described. As shown in our experimental results, after applying the rectification algorithm on the sub-images captured using the single-lens stereovision system, the “slanted” epipolar lines are transformed and become parallel to the horizontal scan lines of the image. In conclusion, we have proven that

the geometry-based approach can be extended to solve the rectification issue of single-lens multi-face prism based stereovision system. The next chapter will present the stereo matching algorithm to solve stereo correspondence problem.

Chapter 5 Segment-based stereo matching using cooperative optimization: image segmentation and initial disparity map acquisition

In chapter 3 and 4, we proposed a geometry-based approach to rectify the virtual cameras which are generated using the single-lens prism based stereovision system. The next important aspect in stereovision is stereo correspondence, which has been widely studied amongst the researchers [14]. The goal of stereo matching is to determine the disparity map between an image pair taken from the same scene. Disparity describes the difference in location of the correspondence pixel and it is often considered as a synonym for inverse depth [127]. The algorithms of stereo matching which are proposed to solve stereo correspondence and obtain the disparity map have been studied thoroughly in the past, but there are still many challenging works needed to be done such as the occurrence of occlusion, scenes with texture-less and/or repeated patterns, and image noise. Current techniques used in stereo matching algorithms are classified [14] into local approaches and global approaches, which have been discussed in Section 2.6.

In this thesis, we propose a segment-based stereo matching algorithm using cooperative optimization to solve stereo correspondence and obtain the disparity map. In our algorithm, firstly, we utilize the mean-shift method [99] to segment the reference image (which can be one of the two captured images). Local matching method (biologically and psychophysically inspired adaptive support weights, BPASW) is then employed to obtain the initial disparity map quickly. Subsequently, we apply the robust disparity plane fitting to segments and use the method of Singular Value Decomposition (SVD) to solve the least square equation. In order to build a set of reliable pixels for the segment, three rules are formulated to filter the outliers. Next, we merge the neighboring segmented regions using the improved clustering

algorithm based on cohesion [116]. Finally, an energy function using cooperative optimization is formulated to obtain the final disparity map. Figure 5.1 shows the flow chart in obtaining the depth map using the proposed stereo matching algorithm mentioned above. We will discuss this algorithm in detail in the next sections.

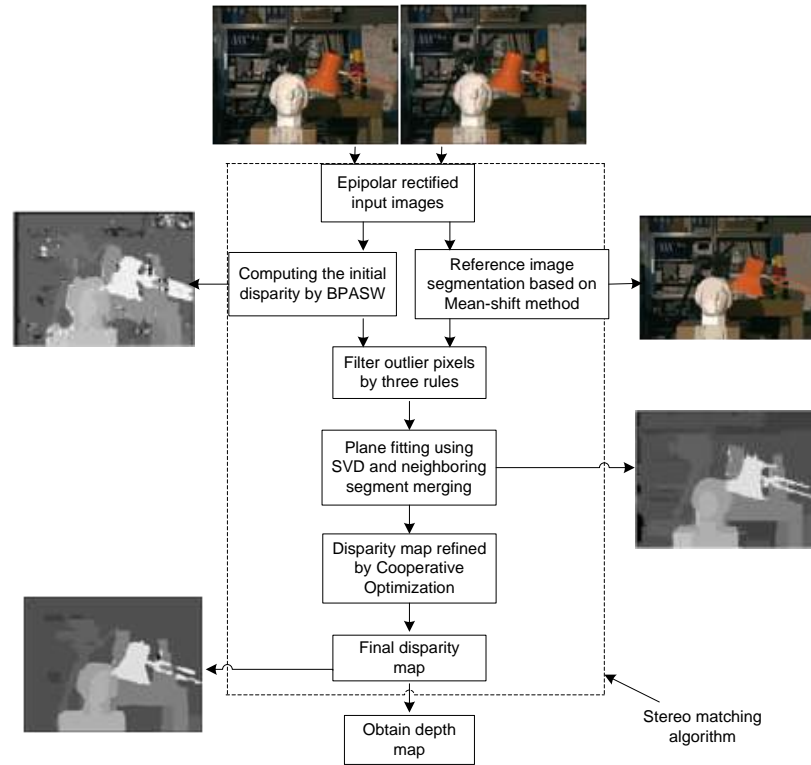


Figure 5.1 The flow chart of obtaining depth map from stereo matching algorithm

This Chapter presents the first two steps of the proposed stereo matching algorithm, namely, image segmentation using mean-shift method and initial disparity map acquisition using BPASW.

5.1 Image segmentation

Image segmentation is the first step of our stereo matching algorithm, which will affect the results of the following steps. In this section, mean-shift method is employed to segment the

reference image. The following sections will present the details on the application of the method. For a review of general image segmentation methods please refer to Appendix C.

5.1.1 Mean-shift method

Mean shift is an unsupervised clustering method [107], which estimates the gradient of a probability density function to detect modes in an iterative fashion. Hence, image segmentations that take color/intensity-similarity as well as local connectivity into account, can be obtained by applying this algorithm to the combined spatial-range domain [99]. The mean shift technique detects modes in a probability density function based on the Parzen Density Estimate [107,108]:

$$\hat{f}_{K_s}(x) = \frac{1}{Nh^n} \sum_{i=1}^N K_s\left(\frac{x - x_i}{h}\right) \quad (5.1)$$

Here, N equals to the number of n -dimensional vectors $x_1 \cdots x_N$. The parameter h is the window radius of the kernel K_s . The multivariate mean shift vector of the point x is given by [99]

$$m_K(x) = \frac{\sum_{i=1}^N x_i K\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^N K\left(\frac{x - x_i}{h}\right)} - x \quad (5.2)$$

For the uniform kernel K_U , the calculation of the mean shift vector (5.2) becomes an average of the vector differences. It can be shown that the mean shift vector is proportional to the normalized density gradient estimate [99]

$$m_K(x) = \frac{1}{2} h^2 c \frac{\nabla \hat{f}_{K_E}(x)}{\hat{f}_{K_U}(x)} \quad (5.3)$$

where c is the corresponding normalization constant and K_E is the radially symmetric Epanechnikov kernel given by

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1 - \|x\|^2), & \|x\| \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (5.4)$$

with c_d being a normalization constant. To ensure the isotropy of the feature space, a uniform color space, such as $L * u * v$ is typically used. In the case of gray value images, the L component of color space is used only. To account for different spatial and tonal variances, it is reasonable to choose a kernel window of size $S_h = S_{(h_s, h_r)}$ with differing radii h_s in the spatial and h_r in the range domains. Since the mean shift vector is designed to be aligned with the local gradient estimate, it can be shown that by successive computation of (5.2) and shifting the kernel window by $m_K(x)$, the mean shift procedure is guaranteed to converge to a point with zero gradient, i.e., to a mode corresponding to the initial position. Modes which are closer than h_s and h_r are grouped together. For segmentation purposes, each pixel is then assigned the color/intensity value of the corresponding mode. Furthermore, regions with less than some pixel count M might be optionally eliminated.

Mean shift segmentation algorithm implementation procedure [99]:

Let x_i and z_i , $i = 1, \dots, n$, be the d -dimensional input and filtered image pixels in the joint spatial-range domain and L_i the label of the i^{th} pixel in the segmented image.

Run the mean shift filtering procedure for the image and store all the information about the d -dimensional convergence point in z_i ;

Delineate in the joint domain the clusters $\{C_p\}_{p=1 \dots m}$ by grouping together all z_i which are closer than h_s and h_r in the range domain;

For each $i = 1, \dots, n$, assign $L_i = \{p | z_i \in C_p\}$;

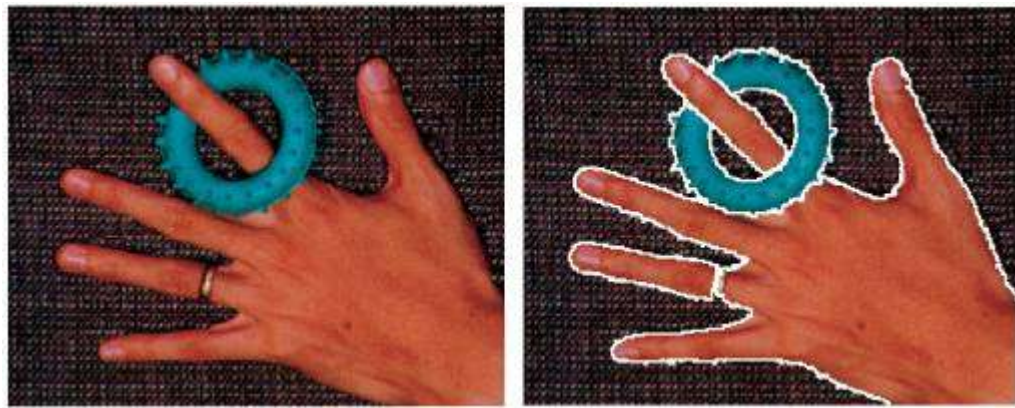
Optional: Eliminate spatial regions containing less than M pixels.

The mean shift procedure is hence an effective algorithm for mode seeking in a density distribution without prior calculation of the distribution itself.

5.1.2 Application of mean-shift method

Mean-shift based segmentation has been successfully applied in several applications [99], including clustering, image segmentation, and filtering [99] with consistent and good results.

Our approach is applied on the segmented region which is obtained through decomposing the reference image into regions of homogeneous color or grayscale. Before the segmentation algorithm, the disparity map is constructed based on two assumptions [109, 110, 78]: (1) disparity values vary smoothly in those regions; and (2) depth discontinuities only occur on region boundaries. In the stereo matching algorithm, disparity continuity within each color segment is assumed. We apply mean-shift color segmentation method which was proposed by Comaniciu and Meer [99]. The mean-shift method is essentially defined a gradient ascent search for maxima in a density function defined over a high dimensional feature space. The feature space includes a combination of the spatial coordinates and all its associated attributes that are considered during the analysis. The main advantage of the mean-shift method is based on the fact that edge information is incorporated that ensures the correctness and precision of our method in large textured-less regions estimation and depth boundaries localization. The segmentation results also satisfy the requirement of our algorithm for computing disparity map in the next steps. The segmentation result is shown in Figure 5.2, and 5.3. The images of hand and room (Figures 5.2 (a) and (c), respectively, are used for testing and the corresponding results are shown in Figures 5.2 (b) and (d). In Figure 5.3 (a), the standard image for testing, Tsukuba has also been segmented using this method; the result is shown in Figure 5.3(b).



(a) Hand image,

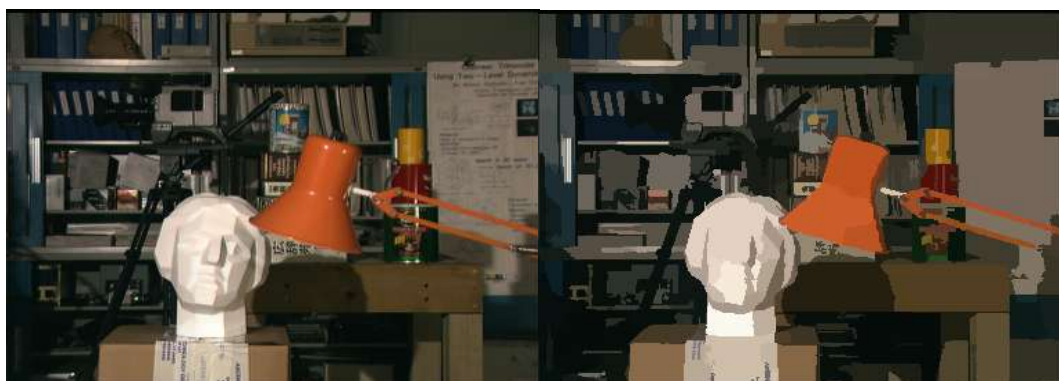
(b) Segmented hand image



(c) Room image,

(d) Segmented room image

Figure 5.2 Segmented by mean-shift method



(a) Reference image

(b) Segmented image

Figure 5.3 Segmented by mean-shift method (using standard image)

5.2 Initial disparity map acquisition

The algorithm of the biologically and psychophysically inspired adaptive support weights (BPASW) is proposed to acquire the initial disparity map in this section. This algorithm will be discussed in detail in the following context.

5.2.1 Biologically inspired aggregation

The success of the human visual system (HVS) in obtaining the depth information from two 2D images still remains an issue and a goal to be accomplished in machine vision [126]. By incorporating the procedures and features from the HVS into artificial stereovision systems, the performance could be improved significantly.

1) Gestalt laws

Gestalt is a movement of psychology that deals with perceptual organization. Gestalt psychology examines the relationships that bond individual elements to form a group [114]. As a consequence, a pattern emerges instead of separate parts. Generally, this pattern exhibits completely different characteristics to its parts.

(1) Gestalt basic grouping principles

Gestalt theory starts with the assumption of active grouping laws in visual perception (Kan97 [112], Wer23 [113]). These groups are identifiable with subsets of the retina. In image analysis we shall identify them as the points of the digital image. Whenever points (or previously formed groups) have one or several characteristics in common, they are grouped to form a new larger visual object, a *Gestalt*. The detail of the gestalt rules by which elements tend to be associated together and interpreted as a group is presented, such as *Vicinity*

(*Proximity*), *Similarity*, *Continuity*, *Common fate*, *Closure*, *Parallelism*, and *Symmetry*. The detailed definitions of group rules are given in Appendix D.

(2) Collaboration of grouping laws

The grouping laws usually collaborate to the objects. In our stereo matching algorithm, the grouping laws of *proximity*, *similarity* and *continuity* are collaborated to get the adaptive support weights.

Aggregation is a crucial stage of a stereo algorithm. Assigning the correct significance weights to each pixel during aggregation is a very difficult decision where gestalt theory can provide a solution [126]. The three aforementioned basic laws take the following meaning

(a) *Proximity (or equivalently Distance)*: The closer two pixels are, the more correlated to each other they are.

(b) *Intensity similarity (or equivalently Intensity dissimilarity)*: The more similar the colors of the two pixels are, the more correlated they are.

(c) *Continuity (or equivalently discontinuity)*: The more similar the depths of two pixels are, the more probable that they belong to the same larger feature and, thus, the more correlated they are.

Thus, collaboration of grouping laws can be used to determine to what degree two pixels are correlated.

2) Psychophysically-based weight assignment

The remaining question is exactly how much a pixel correlated to another should contribute to it during the aggregation process. In other words, it is necessary to establish an appropriate

mapping between correlation degree and contribution [126]. The Weber-Fechner law is one of those theories and is widely acceptable. It indicates a logarithmic correlation between the subjective perceived intensity and the objective stimulus intensity. The mathematical expression of Weber-Fechner law is

$$l = -k \ln \frac{S}{S_0} \quad (5.5)$$

where l is the perceived intensity, S is the stimulus' intensity at the instant and k is a positive constant determined by the nature of the stimulus. S_0 is the stimulus' value that results in zero perception and under which no stimulus' change is noticeable. The detailed expression is given in Appendix D.

5.2.2 Initial disparity map estimation algorithm

In this section, we utilize Gestalt laws and the psychophysically-based weight assignment to compute the initial disparity map.

The self-adaptation dissimilarity measure is used to compute the aggregation. The dissimilarity of the two pixels can be estimated by the Absolute intensity Differences (AD) (section 2.6) of their intensities and a gradient based measure. The Self-Adaptation Dissimilarity function (S_{AD}) is as follows:

$$\begin{aligned} S_{AD}(x, y, d) = & (1 - \omega) |I(x, y) - I'(x + d, y)| \\ & + \omega (|\nabla_x I(x, y) - \nabla_x I'(x + d, y)| \\ & + |\nabla_y I(x, y) - \nabla_y I'(x + d, y)|) \end{aligned} \quad (5.6)$$

where ∇_x and ∇_y represent the gradient map in the $+x$, and $+y$ directions, respectively. ω denotes the weight between the intensity difference and a gradient based measure. d is the disparity value of each pixel.

1) Gestalt law weights computation

Collaboration of gestalt grouping laws is used to determine to what degree two pixels are correlated. Then we use gestalt law weights for psychophysically-based aggregation of the two pixels.

Let (x, y) be the coordinates of the central pixel of support region, (x', y') is the coordinates of a pixel lying inside its support region.

(a) *Proximity*

Proximity of the two pixels is taken into consideration using their Euclidean distance on the image plane. The distance of the pixel (x', y') from the pixel (x, y) is calculated as:

$$\text{dist}(p'(x', y'), p(x, y)) = \sqrt{(x - x')^2 + (y - y')^2} \quad (5.7)$$

(b) *Dissimilarity*

In dissimilarity part, we use AD dissimilarity measure. Thus, the dissimilarity between the pixels (x', y') and (x, y) is calculated as:

$$\text{dissimilarity}(I(x, y), I'(x', y')) = |I(x, y) - I'(x', y')| \quad (5.8)$$

(c) *Discontinuity*

The continuity of two pixels can be described by the possibility that they both have the same depth, i.e. they share the same disparity value. If the disparity of the two pixels is different, it is called discontinuity. The discontinuity between the pixels (x', y') and (x, y) is calculated:

$$discontinuity(x', y', d') | (x, y, d) = \zeta, \quad \text{if } (d \neq d') \quad (5.9)$$

Where ζ is the discontinuity penalty constant. d and d' are the disparities of pixel (x, y) and (x', y') , respectively.

2) Psychophysically based Aggregation

Eq. (5.7)-(5.9) quantify the Gestalt theory. The exact impact of those gestalt laws on weight is obtained by applying the Weber-Fechner law, as described in Eq. (5.5). The factor S_0 of the Weber-Fechner law for this case is equal to unity and can be neglected. Thus, the weighting factor due to each gestalt law, distance, dissimilarity and discontinuity, can be calculated [126]:

$$\omega_{dist}(x', y', d) | (x, y, d) = -k_1 \ln(dist(p'(x', y'), p(x, y))) \quad , \quad (5.10)$$

$$\omega_{dissim}(x', y', d) | (x, y, d) = -k_2 \ln(dissimilarity(I, I')) \quad , \quad (5.11)$$

$$\omega_{dic}(x', y', d) | (x, y, d) = -k_3 \ln(discontinuity(x', y', d)) \quad , \quad (5.12)$$

The total weight is computed by multiplying the three individual weights in Eq. (5.10-5.12):

$$\omega_{tot} = \omega_{dist} \cdot \omega_{dissim} \cdot \omega_{dic} \quad . \quad (5.13)$$

The total weight is calculated for both the left and the right input images according to Eq. (5.13). Both the $\omega_{tot,l}$ and $\omega_{tot,r}$ are obtained respectively. Note that the subscripts, l and r denote that the weights are associated with the left and right images, respectively.

By taking into consideration the weighting factor for each pixel, the aggregation of the S_{AD} (Eq.(5.6)) is performed and the cost function is:

$$C(x, y, d) = \frac{\sum \omega_{tot,l} \cdot \omega_{tot,r} \cdot S_{AD}(x, y, d)}{\sum \omega_{tot,l} \cdot \omega_{tot,r}} \quad (5.14)$$

The choice of values for the constants $-k_i (i = 1, 2, 3)$ is a difficult and ambiguous task. However, the fraction in the previous equation can be reduced and the values of these constants become trivial for the proposed algorithm, since constants $k_i (i = 1, 2, 3)$ are dropped in the cost function Eq. (5.14).

3) Initial disparity Selection

Finally, the best disparity value for each pixel is decided using a simple winner-takes-all decision methodology. For each pixel, the candidate disparity value d that provides the smaller aggregated value is selected as the pixels' disparity. The calculated disparity values for the whole picture comprise the disparity map $D(x, y, d)$:

$$D(x, y, d) = \operatorname{argmin} C(x, y, d) \quad (5.15)$$

The computational load of the proposed algorithm increases proportionally to the square of the window size, as there is no data-dependent iterative process that is required to converge. This is the typical case for any adaptive support weight (ASW) stereo algorithm [115].

5.3 Experimental results and discussion

In this section, the results of initial disparity map acquisition are presented. The biologically and psychophysically inspired adaptive support weights algorithm are proposed to compute the initial disparity map.

5.3.1 Experimental procedure

We present an efficient algorithm to obtain the initial disparity map. The procedure of the initial disparity map acquisition is shown in Figure 5.4 [126]. The accuracy of results is obtained after implementation this algorithm, the detail results is shown in section 5.4.2.

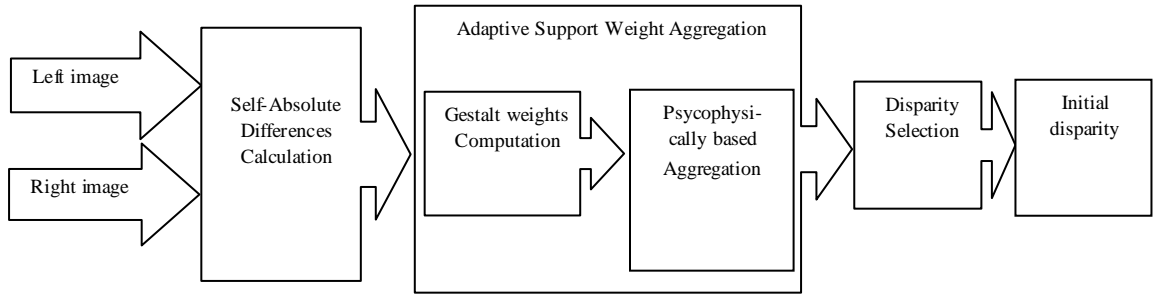


Figure 5.4 Block diagram of the algorithm's structure

5.3.2 Experimentation results

We use the pairs of standard images, “Tsukuba, Cones, Teddy, and Venus” with their corresponding ground-truth maps to validate our algorithm. The ground-truth maps which are provided by Middlebury website [14], are considered as the “model” disparity maps. The disparity maps which are extracted from standard images by proposed algorithms are compared with the ground-truth maps. The Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD), Normalized Cross Correlation (NCC), Sum of Hamming Distance (SHD) and our method are applied to obtain the initial disparity map for comparison. Figure 5.5 shows the results of these methods:

Row 1: Reference images (Tsukuba, ones, Teddy, and Venus);

Row 2: the standard images’ corresponding ground-truth maps;

Row 3: Disparity map by SAD;

Row 4: Disparity map by SSD;

Row 5: Disparity map by NCC;

Row 6: Disparity map by SHD;

Row 7: Disparity map by our method.



Figure 5.5 Initial disparity maps by five methods (SAD, SSD, NCC, SHD, our method)

5.3.3 Analysis of results





By comparing the disparity map obtained by the local stereo matching algorithms with the ground-truth map, the percentage of bad matching pixels (or percentage of bad corresponding points) [14] can be computed by Eq.(5.16) ,

$$B = \frac{1}{N} \sum_{(x,y)} (|d_C(x,y) - d_T(x,y)| > \delta_d) \quad (5.16)$$

where B is the percentage of bad matching pixels, N is the total number of pixels, $d_C(x,y)$ is the computed disparity map, $d_T(x,y)$ is the ground-truth map and the threshold δ_d is defined to be 1.0 based on the literature [14].

The initial disparity map results using the five methods are shown in Figure 5.6. Table 5.1 shows the percentages of the bad matching pixels of the reference image. By comparing the average percentages of the bad matching pixels, our method results in the lowest percentage of bad matching, followed by the NCC; and the SHD yields the highest percentage of bad matching amongst all the methods. The results of disparity maps extracted from the whole reference image, which includes textureless, discontinuity and occlusion part, are more accurate by proposed method. This shows that our method can obtain a more accurate initial disparity map which is better for the disparity plane estimation which will be shown in the next Chapter. It is also verified that the biologically inspired aggregation method works best in this study. Because we calculate the degree of correlation between neighbor pixel and center pixel in the support region (Section 5.2.2), we then use the psychophysical principle to compute the contribution of each of degrees of correlation (Eq. (5.10-5.12)). The end result would be a more accurate relationship between the centre pixel and its neighbors in the support region. Therefore, our method can be used to obtain the initial disparity map for the reference images.

Table 5.1 Percentages of bad matching pixels of reference images by five methods

Percentage of bad matching pixels Methods	Reference images Tsukuba 	Cones 	Teddy 	Venus 	Average
SAD	26.71	18.02	17.04	20.36	18.03
SSD	23.474	16.04	16.38	11.06	16.74
NCC	22.37	15.21	12.38	9.56	14.88
SHD	31.72	26.53	24.62	15.78	24.66
Our method	19.49	16.03	11.36	5.08	12.99

5.4 Summary

This chapter presents the two initial steps of our proposed segment-based stereo matching algorithm using cooperative optimization, they are: image segmentation and initial disparity map acquisition. In Section 5.1, the method of mean-shift, which is based on the fact that edge information is incorporated that ensures the correctness and precision of our proposed algorithm in large texture-less regions estimation and depth boundaries localization, is employed for image segmentation to satisfy the requirement for disparity map computation in the later steps. In Section 5.2, the biologically and psychophysically inspired adaptive support weights algorithm (BPASW) is proposed to acquire the initial disparity map, in which the Gestalt laws and the psychophysically-based weight assignment are utilized and the exact impact of these Gestalt laws on weight is obtained by applying the Weber-Fechner law. By comparing the disparity map obtained by other local stereo matching algorithms (SAD, SSD, NCC, SHD) with the ground-truth map, the proposed method yields better results based on our experiment results. This verifies the effectiveness of our proposed algorithm (BPASW). Besides, this will also improve the overall accuracy of our matching algorithm when the

subsequent steps is executed based on the initial disparity map. The next Chapter will introduce the rest of the steps of the stereo matching algorithm.

Chapter 6 Segment-based stereo matching using cooperative optimization: disparity plane estimation and cooperative optimization for energy function

In Chapter 5, we have described the first two steps of the stereo matching algorithm, which are image segmentation and initial disparity map acquisitions. The remaining steps of the proposed stereo matching algorithm such as outlier filtering, plane fitting and neighboring segmentation merging, and final disparity map acquisition, are presented in this chapter. The structure of this chapter is separated mainly into two parts, namely, disparity plane estimation and cooperative optimization of energy function. The first part includes plane fitting by Singular Value Decomposition (SVD), filtering outliers by three rules, and merging of neighboring disparity planes by improved clustering algorithm. The second part includes the cooperative optimization algorithm and the formulation of energy function.

6.1 Disparity plane estimation

In our approach, obtaining regions from color image segmentation is essentially the first step of the proposed stereo matching algorithm. Based on the assumption that no large depth discontinuities exist inside a color segment, the depth of the pixels in a segment can be modeled using some representations [109]. Thus, fitting the disparity plane becomes a significant step in segment-based stereo matching algorithm. In order to obtain a more accurate disparity plane, robust plane fitting method is applied to fit the segmented regions and simple measurement to merge neighboring disparity plane. The detailed implementation of the methodology is presented in the subsequent sections.

6.1.1 Plane fitting

The purpose of plane fitting is to determine the disparity plane of a segment using initial disparity value. Tao et al. [109] presented a detailed description of this process.

The disparity plane of a segment is modeled as

$$d^j(x, y) = ax + by + c \quad (6.1)$$

where a, b and c are the plane parameters of j^{th} segment and $d^j(x, y)$ is the disparity plane of the j^{th} segment, (x, y) is the coordinate of pixel in the j^{th} segment. Now, if there are n segments in the reference image, there will be n disparity planes.

The parameters (a, b, c) are obtained from the least squares solution of a linear system as shown in Eq. (6.2)

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_p & y_p & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} d_1^j \\ d_2^j \\ \vdots \\ d_p^j \end{bmatrix} \quad (6.2)$$

$$\text{Let } A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_p & y_p & 1 \end{bmatrix} \text{ and } B = \begin{bmatrix} d_1^j \\ d_2^j \\ \vdots \\ d_p^j \end{bmatrix}$$

where p is the number of pixels in j^{th} segment, the i^{th} row of A is $[x_i, y_i, 1]$ and the i^{th} element in B is $d^j(x_i, y_i)$.

Eq. (6.3) can be derived from Eq. (6.2) by multiplying A^T on both sides of the equal sign.

$$A^T A [a, b, c]^T = A^T B \quad (6.3)$$

Expanding Eq. (6.3) gives:

$$\begin{bmatrix} \sum_{i=1}^p x_i^2 & \sum_{i=1}^p x_i y_i & \sum_{i=1}^p x_i \\ \sum_{i=1}^p x_i y_i & \sum_{i=1}^p y_i^2 & \sum_{i=1}^p y_i \\ \sum_{i=1}^p x_i & \sum_{i=1}^p y_i & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^p x_i d_i^j \\ \sum_{i=1}^p y_i d_i^j \\ \sum_{i=1}^p d_i^j \end{bmatrix} \quad (6.4)$$

Next, we propose using the Singular Value Decomposition (SVD) method to obtain the least square solution.

$$[a, b, c]^T = (A^T A)^+ A^T B \quad (6.5)$$

where $(A^T A)^+$ is the pseudo-inverse of $A^T A$, $(A^T A)^+$ which can be computed using SVD.

$$SVD(A^T A) = U D V^T = [u_1, u_2, \dots, u_m] \begin{bmatrix} D_0 & 0 \\ 0 & 0 \end{bmatrix} [v_1, v_2, \dots, v_n]^T$$

$$(A^T A)^+ = [v_1, v_2, \dots, v_m] \begin{bmatrix} D_0^{-1} & 0 \\ 0 & 0 \end{bmatrix} [u_1, u_2, \dots, u_n]^T \quad (6.6)$$

where vector u_i is the eigenvector of $A^T A (A^T A)^T = (A^T A)^2$, and

$$D_0 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r), \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0,$$

σ_i^2 are the nonvanishing eigenvalues of $(A^T A)^2$, or

σ_i are the nonvanishing singular values of $A^T A$, and

vector v_i is the eigenvector of $(A^T A)^T A^T A = (A^T A)^2$.

From the definition of u_i and v_i , we know that the vector $u_i = v_i$. Due to this property, the complexity of the least square problem is reduced.

There are several advantages using SVD in solving the least square equation. First, considering the case where only the matrix A is known, the unknown parameters can be obtained using SVD. Besides, SVD allows the calculation of the psedo-inverse of $(A^T A)^+$ through SVD with no regard to the singularity of $A^T A$.

6.1.2 Outlier filtering

The fitting process should be robust in filtering outliers. As the least square solutions are very sensitive to outliers, the estimated plane might be disturbed due to the remaining outliers. These outliers will reduce the accuracy of disparity fitting results. Therefore, rules shall be formulated to filter them. We implement three rules for the said purpose: cross-checking, assessing reliable segment, and measuring the distance between the previously determined initial disparity to the current computed disparity. The following sections will show the detailed descriptions of the rules.

1) Cross-checking.

Cross-checking is adopted to obtain the reliable pixels and filter out the occluded pixels and areas with low texture details where disparity estimation tends to be unreliable. Let D_L be the disparity set which is obtained from matching the left image to the right image and D_R be the disparity set by matching the right image to the left image.

Cross checking condition is expressed as shown below:

$$|D_L(x_L) - D_R(x_L - D_L(x_L))| < 1 \quad (6.7)$$

If the pixel's disparity satisfies Eq. (6.7), we consider the pixel to be a reliable pixel and vice versa.

The reliable set $U' = \{(x_i, y_i) \in U \mid |D_L(x_L) - D_R(x_L - D_L(x_L))| < 1\}$.

where U is the set of all pixels inside the segment.

2) Assessing reliable segment.

We build a rule to judge whether a segment is reliable or unreliable, the regularity is defined as follows:

$$\rho_1 = \frac{N_{unreliable}}{N_{segment}} > \gamma_{segment} \quad (6.8)$$

where ρ_1 is the ratio between the number of the unreliable pixels in the segment, $N_{unreliable}$, and the total number of the pixels in the segment, $N_{segment}$. $\gamma_{segment}$ a constant, is a pre-defined threshold.

If a segment satisfies the criterion in Eq. (6.8), all the pixels in the segment are then labeled as unreliable, which indicates the lack of sufficient data to provide reliable plane estimations, and vice versa. These segments are ignored.

Thus, the new reliable pixel set is

$$U'' = \begin{cases} U', & \text{if } \{(x_i, y_i) \in U \mid \frac{N_{unreliable}}{N_{segment}} \leq \gamma_{segment}\} \\ \emptyset, & \text{or else} \end{cases}.$$

3) Measuring the distance between previously determined initial disparity to the current computed disparity.

The distance between the previously determined initial disparity to the current computed disparity is measured, and the rule is stated as follows:

$$|d_k^j - (ax_k + by_k + c)| \leq t_{outlier} \quad (6.9)$$

where d_k^j is the initial disparity of pixel (x_k, y_k) in j^{th} segment (see Section 5.2), and for every disparity of pixel in the j^{th} segment, we evaluate using $ax_k + by_k + c$, for $k = 1$ to p , where p is the total number of pixel in the j^{th} segment. $t_{outlier}$ is a outlier threshold.

Therefore, the reliable set is updated to U''' .

$$U''' = \{(x_k, y_k) \in U'' \mid |d_k^j - (ax_k + by_k + c)| \leq t_{outlier}\}$$

If a segment satisfies the condition stipulated in Eq. (6.9), then the all the pixels in the segment are labeled as reliable and vice versa.

After removing the outliers, the plane fitting procedure given in Section 6.1 is executed to find the new parameters of the disparity plane a' , b' and c' . The convergence criterion in Eq. (6.10)

$$e^{-(|a-a'|+|b-b'|+|c-c'|)} > \varepsilon \quad (6.10)$$

is defined. Note that in Eq. (6.10), a' , b' and c' are the new parameters of the plane; a , b and c are the parameters of the plane that was obtained in the previous iteration, and ε is threshold of the convergence value (typically 0.99). The plane fitting procedure is executed until convergence criterion shown in Eq. (6.10) is satisfied. Figure 6.1 shows the detailed algorithm of the estimated disparity plane parameters. The detailed implementation of the algorithm is as follows:

Input segmented image which contains n segments; // where n is the number of segments in //the image

Obtain initial disparity map set $D(x, y)$;

For ($i = 0$; $i < n$; $i++$)

Cross-checking;

If ($\rho_1 = \frac{N_{unreliable}}{N_{segment}} > \gamma_{segment}$)

Calculate disparity parameters by *SVD*;

If ($|d_k^j - (ax_k + by_k + c)| \leq t_{outlier}$)

Get the disparity parameters a', b', c' ;

If ($e^{-(|a-a'|+|b-b'|+|c-c'|)} > \epsilon$)

Obtain the segment current disparity parameters;

Else

Calculate the disparity using current disparity parameters;

End if

Else

Update the reliable pixels set;

End if

End if

End for

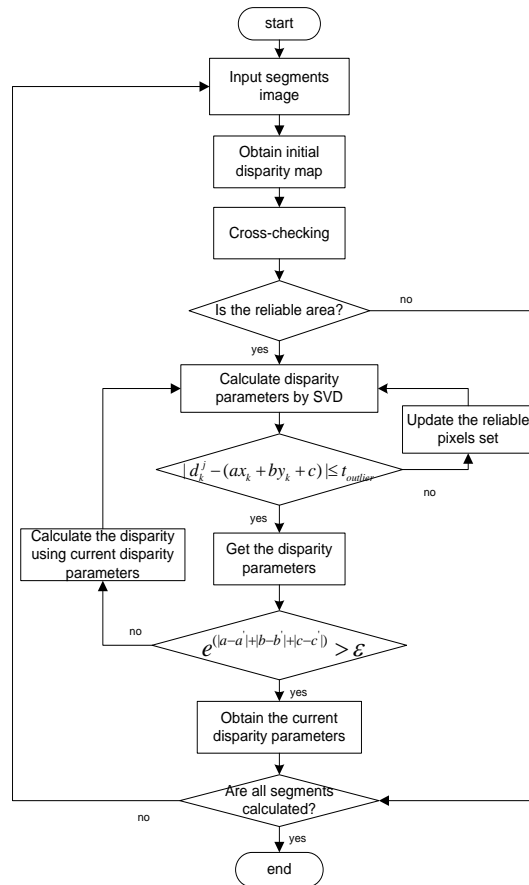


Figure 6.1 The flow chart of the estimated disparity plane parameters

6.1.3 Merging of neighboring disparity planes

The relationship between the neighboring disparity planes is investigated after fitting the disparity plane. As one single surface that contains texture is usually divided into several segments after applying the mean-shift method, merging of those segments is important for the estimation of the disparity plane parameters. In this case, we utilize the clustering algorithm [116] and define the rules on similarity measure for clustering. The geometrical relationship of the adjacent planes, such as parallelism and intersection are employed to determine the criteria of whether two planes should be merged. The definitions of parallelism and intersection of the adjacent planes are shown in the following sections.

1) Segment representation

In our algorithm, the segment s_j represents the j^{th} segment and its correspondence disparity plane is $p_j \equiv \{a_j x + b_j y + c_j = d^j(x, y)\}$, where a_j, b_j and c_j are the parameters of the disparity plane. After segmentation, the definition of neighboring segments set is

$$\mathcal{N}_j = \left\{ s_i \mid \bigcup_{i=1}^n s_i \text{ and } s_i \cap s_j \neq \emptyset, i \neq j \right\} \quad i, j = 1, \dots, q \quad (6.11)$$

where \mathcal{N}_j is the j^{th} neighboring segments set, and s_i is a neighboring segment of s_j . Comparison between s_i and s_j is done by using the rule of similarity measures.

2) Similarity measurement

The similarity between two segments is measured by calculating the similarity between the disparity planes. Properties of planes, parallelism and intersection are used, in the similarity measurements.

Given two segments, segments A and B randomly, the plane equations are given by:

$$d_A = a_A x + b_A y + c_A \quad (6.12)$$

$$d_B = a_B x + b_B y + c_B \quad (6.13)$$

Then $v_1 = [a_A \ b_A \ c_A]^T$, and $v_2 = [a_B \ b_B \ c_B]^T$ are the respective normal directional vectors of the planes A and B, respectively. The decision of whether two planes can be merged is determined by their properties including intersection and parallelism.

The two rules are designed as follows:

(1) In the case of two intersecting planes, we compute the angle between the neighboring disparity planes using

$$\theta = \arccos \frac{(v_1^T v_2)}{\sqrt{(v_1^T v_1)(v_2^T v_2)}} \quad (6.14)$$

where θ is the angle of intersection of the two neighboring planes.

There are the two types of relationship between the disparity planes, $\theta \leq \frac{\pi}{2}$, and $\theta > \frac{\pi}{2}$, which are shown in Figure 6.2(a), and (b).

(2) In the case of two parallel planes, we compute the distance between the neighboring segment planes using

$$d_{dist} = \frac{|d_A - \frac{d_B}{\tau}|}{\sqrt{(v_1^T v_1)}} \quad (6.15)$$

where τ is the scale factor which transforms Eq.(6.12) to Eq.(6.13) with the same disparity plane parameters. Figure 6.2(c) shows the distance between two disparity planes.

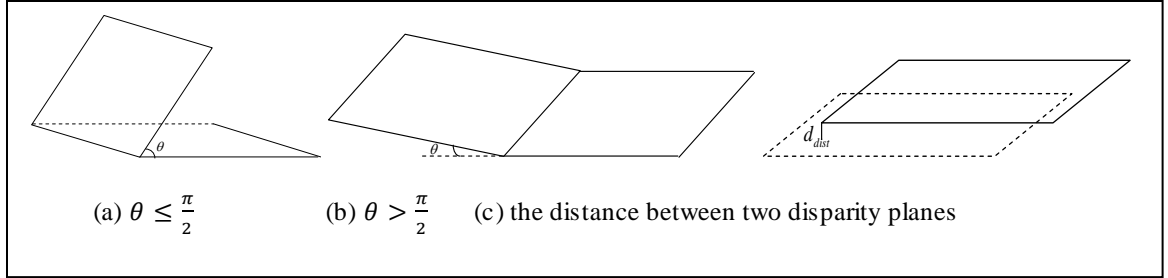


Figure 6.2 Two type properties of plane

From the above development, we define the similarity function for the neighboring disparity planes, as follows:

$$simfun = \begin{cases} e^{-\theta} , & \text{if } A \nparallel B, \theta \leq \frac{\pi}{2} \\ e^{-(\pi-\theta)} , & \text{if } A \nparallel B, \theta > \frac{\pi}{2} \\ e^{-d_{dist}} , & \text{if } A \parallel B \end{cases} \quad (6.16)$$

where *simfun* denotes the similarity function.

The decision on whether to merge the two neighboring planes is based on the following criterion:

$$simfun > \delta \quad (6.17)$$

where δ is a constant threshold.

3) Clustering procedure

The designed similarity function (Eq. (6.16)) is then used to cluster segments. Note that a segment is represented by its disparity plane. Hence, clustering of segments can be viewed in the merging of disparity planes. The details of clustering procedure are described below:

Let q be the number of neighboring segments of s_j , where s_j is a segment. The procedure of the clustering algorithm is given below:

- 1) The q segments in the set are placed in q distinct clusters. These clusters are indexed by $C \equiv \{C_i | i = 1, 2, \dots, q, i \neq j\}$. Cluster C_j represents segment s_j , we shall compute the similarity function (Eq.(6.16)) between C_j and the neighboring clusters.
- 2) A cluster C_l ($l = 1, 2, \dots, q, l \neq j$), which the similarity function (Eq.(6.16)) is carried out with C_k is selected when the condition stated in Eq.(6.17) is satisfied. These two clusters are merged into a new cluster.
- 3) Recursively performing steps 2 with another cluster and 3 until the number of clusters has reduced to a required number or all the clusters have been visited.

We can merge the similarity planes by implementing the above clustering algorithm. Figure 6.3 shows the flow chart for the procedure of merging neighboring disparity planes. The implementation of merging neighboring disparity planes is shown in detail below:

```

Input set of fitting plane  $P_k(a, b, c), k = 1, \dots, n;$  // where  $n$  is the number of segments in
//the image
For ( $i = 0; i < n; i++$ )
     $P_i(a, b, c);$  // randomly select one plane
    For ( $j = 0; j < m; j++$ ) // where  $m$  is the number of the neighboring segments of  $S_i$ 
        // where  $S_i$  is the correspondence segment of  $P_i(a, b, c)$  plane
        Compute the angle of the  $S_i$  and its neighboring segment  $S_j$ ;
        Compute the distance of the  $S_i$  and its neighboring segment  $S_j$ ;
        If ( $simfun > \delta$ )
            Merge the two segments;
            Compute the plane parameter again using SVD;
        End if
    End for
End for

```

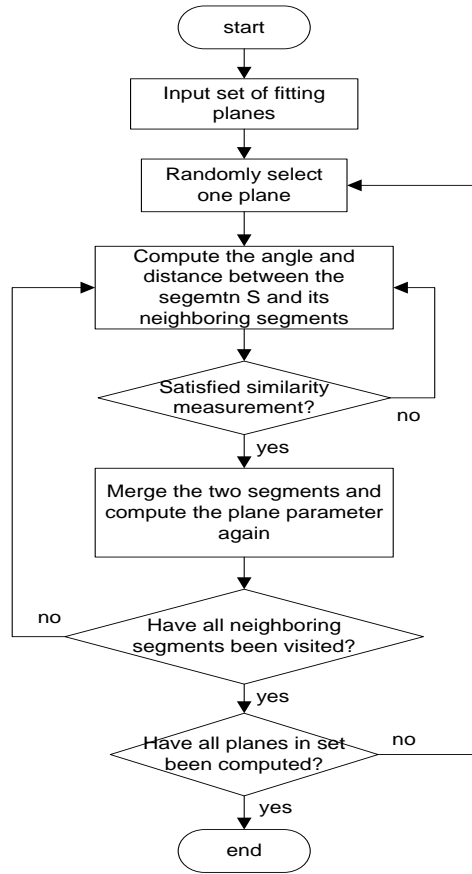


Figure 6.3 The flow chart for the procedure of merging the neighboring disparity plane

6.1.4 Experiment

The method of disparity plane estimation which includes plane fitting and the neighboring segment merging has been described above. In this section, we shall present the experimental results to demonstrate their effectiveness.

1) Experimental results

The results of disparity plane estimation are shown in Figure 6.4. Standard images, “Tsukuba, Cones, Teddy, and Venus” with their corresponding ground-truth maps are used in the experiment. In Figure 6.4, 1st row shows the reference images (Tsukuba, Cones, Teddy, and Venus), 2nd row presents the ground-truth maps (Section 5.3); 3rd row is the disparity map

obtained by the method of initial disparity map acquisition (Section 5.2), and the 4th row is the disparity map obtained by the method of disparity plane estimation (Section 6.1).

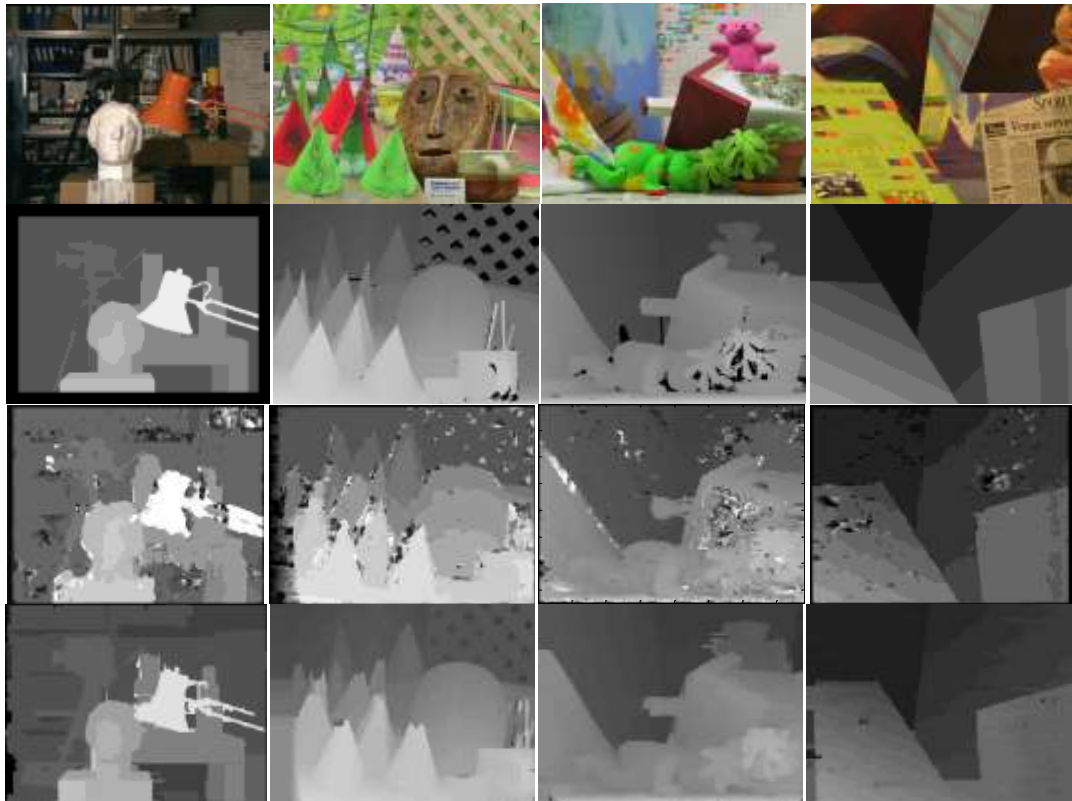






Figure 6.4 The results of disparity map obtained in each stage

2) Analysis of results

The disparity maps obtained by the method of disparity plane estimation are compared with the ground-truth maps using the percentage of bad matching pixels which has been defined in section 5.3.3 (Eq. (5.16)). Table 6.1 shows the percentage of bad matching pixels of both the disparity maps. One of them, the initial disparity map has been obtained in Section 5.3.3, and the other one is determined by our proposed method in Section 6.2. By comparing the average percentages of bad matching pixels of both the methods, the proposed method achieves a lower percentage of bad matching pixels. In other words, this method is comparatively better in obtaining the disparity maps.

Table 6.1 Percentages of bad matching pixels of disparity map obtained by the two methods compared with ground-truth

Percentage of bad matching pixels / Methods	Reference images	Tsukuba	Cones	Teddy	Venus	Average
Initial disparity map						
		19.49	16.03	11.36	5.08	12.99
Disparity map by our proposed method		5.48	8.35	7.42	2.37	5.91

6.2 Cooperative optimization of energy function

After presenting the three major steps of the proposed algorithm for obtaining the disparity map, namely, segmentation, initial disparity estimation and the disparity plane estimation. In this section, we describe the formulation of the stereo matching problem as an energy minimization process in the segment domain. The objective is to refine the disparity values.

6.2.1 Cooperative optimization algorithm

In conventional cooperative optimization [117], a complex target is decomposed into several comparatively simple sub-targets, and these sub-targets are optimized individually by taking into consideration the influence of the neighboring sub-targets. In our case, the objective of cooperative optimization is to optimize the disparity plane parameters (a , b , and c) of a segment together with the neighboring segments.

As shown in Figure 6.5, $s_1, s_2, s_3, \dots, s_n$ are the segments acquired by the segmentation method (Section 5.1). Let $E(x)$ represent the total energy function of all segments, E_j represents the j^{th} segment's energy function, the cooperative optimization algorithm will decompose it into the sum of the individual segments' energy function as follows:

$$E(x) = \sum_{j=1}^n E_j \quad (6.18)$$

where, E_j , $j = 1, 2, \dots, n$, is the energy function of the j^{th} segment s_j . This method converts the disparity computation problem to a segment based optimization problem with multiple segments. It is obvious that if we minimize the individual energy function solely for each segment, the final optimization results may not be correct because the influence of the neighboring segments are not considered.

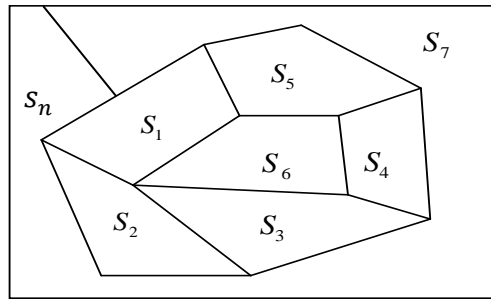


Figure 6.5 Segments after implementation of mean-shift method

Therefore, in order to obtain accurate disparities, minimization of the total (all the sub-targets) energy function is necessary, which is implemented by using cooperative optimization. The solution minimizes all the energy function of a segment and its adjacent segments simultaneously, and then propagates the results via iterative computation. The above optimization process is carried out iteratively until the algorithm converges.

For the iterative process of a segment, E_i , which is the energy of the segment s_j is included in the Eq. (6.18). The neighbor segments s_1, s_2, \dots, s_n of s_j are taken into account. The energy of segment s_j is related to the energy of the neighboring segments in the following expression:

$$(1 - \kappa_j)E_j(x) + \kappa_j \sum_{i \neq j} \mu_{ji} E_i(x) \quad i, j = 1 \dots n \quad (6.19)$$

where, $E_i(x)$ is the energy of the i^{th} segment, s_i ($i = 1, 2, \dots, n; i \neq j$) is an adjacent segment of s_j ; $0 \leq \kappa_j \leq 1$, and $0 \leq \mu_{ji} \leq 1$ are the corresponding weights. By referring to Figure 6.5, considering segment s_6 and optimization of Eq. (6.19), the values of j and i are $j = 6$ and $i = 1, 2, 3, 4, 5$, respectively.

6.2.2 The formulation of energy function

After establishing the relationship of energy function between segments, we then define the energy function to obtain the optimized disparity plane parameters.

The energy function minimization is computed for each segment $s_j \in R$, where R is the segmented image. Corresponding plane parameters $(a_j, b_j, c_j) \in D$, where D is the disparity plane set is obtained from Eq. (6.4) in Section 6.1. The energy function for the disparity plane parameters $(a_j, b_j$ and $c_j)$ is given by:

$$E_j = E_{j_{data}} + E_{j_{occ}} + E_{j_{sm}} \quad (6.20)$$

where E_j is the energy of j^{th} segment, $E_{j_{data}}$ is the j^{th} segment data cost, $E_{j_{occ}}$ is the j^{th} segment's occlusion pixel penalty function and $E_{j_{sm}}$ is the j^{th} segment smoothness function.

For the data term in the cost function, the matching cost is calculated for each disparity plane fitting task. It is computed by summing up the matching cost for each pixel inside the segment s_j , the function is as follows:

$$E_{j_{data}} = \sum_{(x,y) \in \mathcal{N}_r} c_{data}(x, y, d) \quad (6.21)$$

where the $c_{data}(x, y, d)$ is defined in Section 5.3.2, (x, y) is the coordinate of pixel in the s_j and d is the disparity of pixel (x, y) in s_j , which can be computed by disparity plane parameters (Eq.(6.1)), \mathcal{N}_r is a set of reliable pixel in s_j segment.

The occlusion term in the energy function is

$$E_{j_{occ}} = \omega_{occ} \cdot N_{occ} \quad (6.22)$$

where ω_{occ} is the penalty coefficient for occlusion, N_{occ} is the number of occluded pixels in the j^{th} segment, s_j (see section 6.1.2).

The smoothness term in the energy function is

$$E_{j_{sm}} = \sum_{(x,y) \in \mathcal{N}_r} s_{sm}(x, y, d) \quad (6.23)$$

$$s_{sm}(x, y, d) = \begin{cases} \gamma \sum_{((x,y),(x',y')) \in \mathcal{N}} \text{dis}((x,y),(x',y'))^{-1} \exp \left[-\beta (I_{(x,y)} - I_{(x',y')})^2 \right], & \text{if } |d^j(x,y) - d'^j(x',y')| \geq 1 \\ 0, & \text{or else} \end{cases} \quad (6.24)$$

where (x', y') is a neighboring pixel of (x, y) . d and d' are the disparities of the pixel (x, y) and (x', y') in s_j , respectively. $I_{(x,y)}$ and $I_{(x',y')}$ are the intensities of pixel (x, y) and (x', y') in s_j , respectively. \mathcal{N} is the set of pairs of neighboring pixels, and $\text{dis}(\cdot)$ is the Euclidean distance of neighbouring pixels. γ is a constant. This energy function encourages coherence in regions of similar color or grey-level. The constant β is chosen to be:

$$\beta = (2 < (I_{(x,y)} - I_{(x',y')})^2 >)^{-1} \quad (6.25)$$

where $\langle \cdot \rangle$ denotes the expectation over current support region [118]. This choice of β ensures that the exponential term in Eq. (6.25) switches appropriately between high and low contrast.

From the definition of Eq. (6.19), the smaller the total energy of the current region, the better the corresponding disparity estimation can be achieved. Here, Powell's method is used to optimize this energy function (refer to Appendix E).

6.2.3 Experiment

In this section, we also use four standard image pairs, together with their corresponding ground-truth maps to evaluate the performance of the cooperative optimization method.

1) Results of cooperative optimization

Figure 6.6 shows the results of our approach. The first and second columns are the reference images and the ground-truth disparity maps, respectively. The third column shows our experimental results.

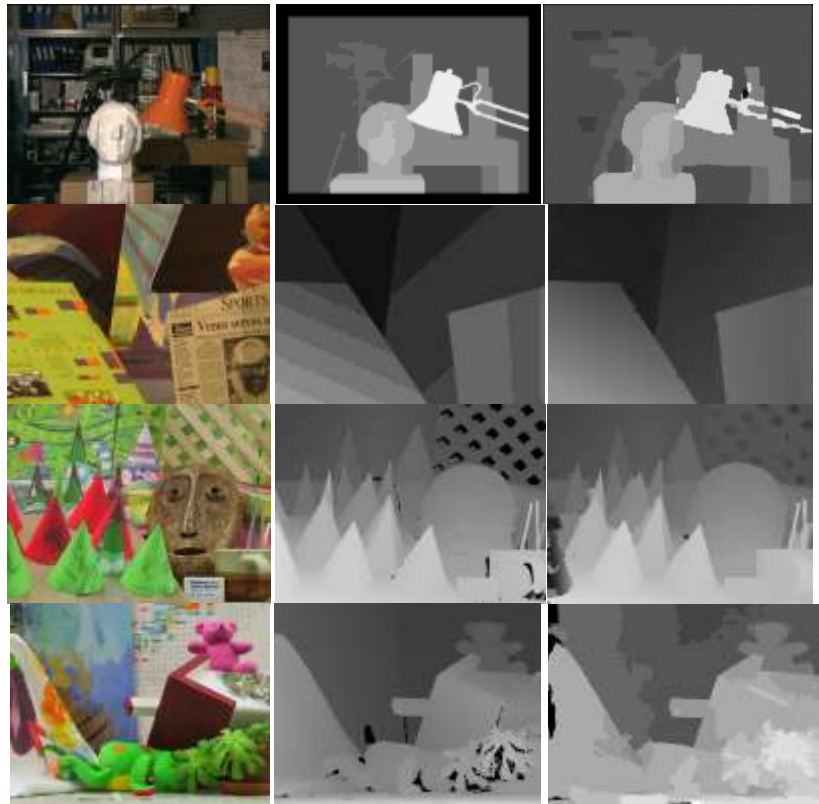
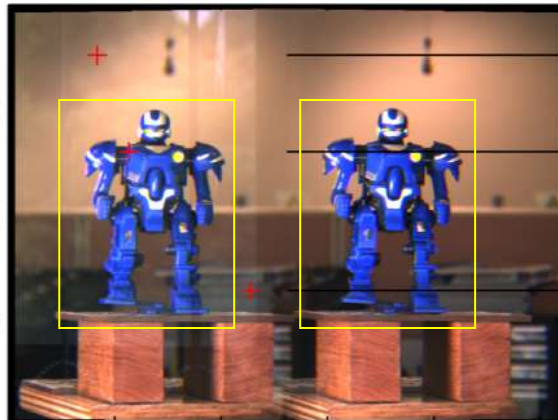


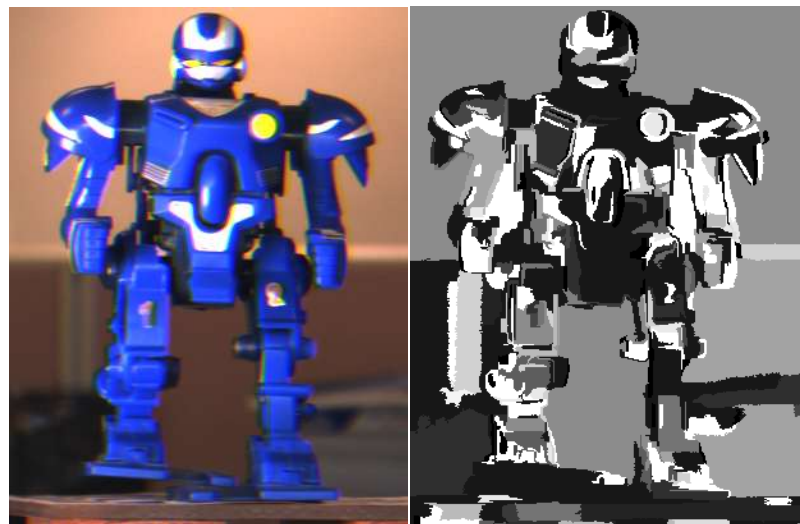
Figure 6.6 Final results of the disparity maps obtained by our algorithm (cooperative optimization)

Next, the images captured from a single-lens stereovision system, “Robot, Pet and Fan”, are used to verify the proposed algorithm. The rectified images (after the implementation of the proposed rectification algorithm in Chapter 3), are shown in Figures 6.7(a), 6.8(a). The regions of interest are extracted as shown in squares in Figures 6.7 (a) and 6.8(b) and the resultant images are shown in Figures 6.7 (b), 6.8(b) and 6.9(a). Then, their corresponding disparity maps are obtained using the proposed algorithm as shown in Figures 6.7(c), 6.8(c) and 6.9(b).

Rectified left image and right image



(a)



(b)

(c)

Figure 6.7 “Robot” images: (a) Rectified image pair, (b) Robot image, which are extracted from rectified image in square, and (c) disparity map

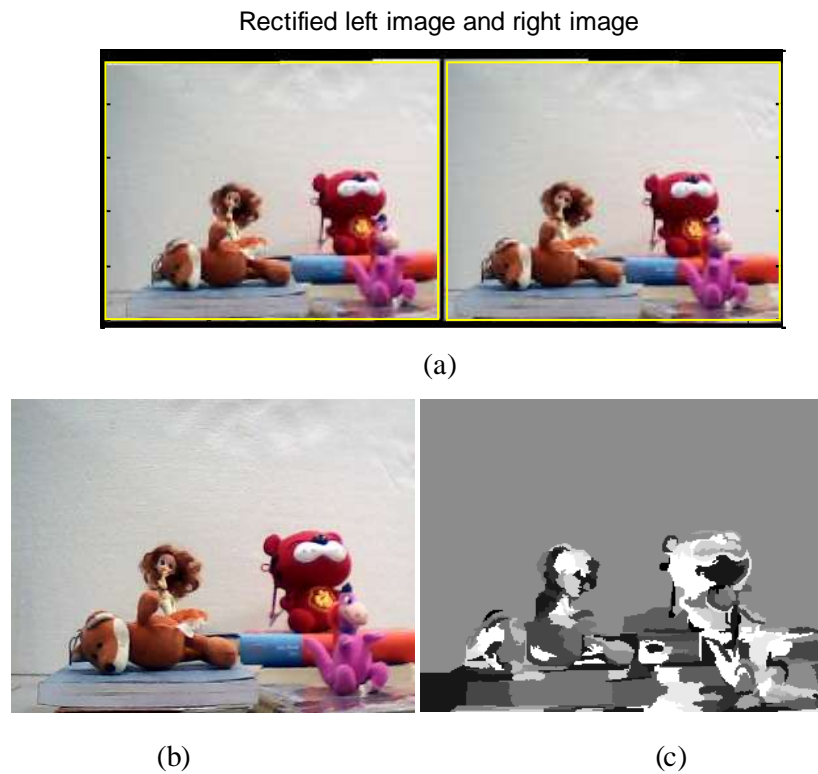


Figure 6.8 “Pet” images: (a) rectified image pair (b) Pet image, which are extracted from rectified image in square, and (c) disparity map



Figure 6.9 “Fan” image: (a) “Fan” image and (b) disparity map

2) Analysis of results

Table 6.2 shows our results using standard test available in Middlebury stereo evaluation website [14]. Our proposed method, with a rank of 3.08, is ranked second amongst the

methods tested. The proposed algorithm produces good results in the challenging areas such as texture-less regions and occluded portions as shown in Table 6.2. Besides, the images (Figures 6.7, 6.8 and 6.9) which are captured using the single-lens prism based stereovision system are used to verify the proposed algorithm. Due to the lack of ground-truth map, the evaluation of the computed disparity map with regard to the ground-truth map is not available at this stage. However, the resultant disparity map can be evaluated using an alternative manner by computing the depth of scene based on the computed disparity map which will be introduced in the next chapter.

The running time of the algorithm is related to the number of iterations. By using a computer with CPU of 2.83 GHz, the total time for the processing of the image pair is about 105s.

The parameters used in the experiments are: $\omega_{occ} = 5$, $\xi_{disc} = 5$, $\kappa_i = 0.5$, and $\gamma_{segment} = 0.9$. In addition, μ_{ij} are set according to ([117]). The constant $\gamma = 50$ has been proven to be a versatile setting for a wide variety of images (see [118]).

Table 6.2 Middlebury stereo evaluations on different algorithms, ordered according to their overall performance

Algorithm	Avg. Rank	Tsukuba Ground truth			Venus Ground truth			Cones Ground truth			Teddy Ground truth		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
AdaptingBP	2.75	1.114	1.375	5.793	0.101	0.211	1.441	2.481	7.923	7.322	4.224	7.063	11.85
Our method	3.08	0.892	1.211	5.844	0.206	0.567	1.722	2.733	6.892	7.041	4.023	6.512	11.24
DoubleBP	3.33	0.881	1.292	4.761	0.134	0.454	1.877	2.905	8.785	7.794	3.532	8.304	9.631
SubPix DoubleBP	6.16	1.2410	1.7611	5.987	0.123	0.465	1.744	2.936	8.739	7.917	3.451	8.386	10.02
GC+Segm Border	9.3	1.4715	1.8212	7.8620	0.195	0.312	2.4410	4.9924	5.781	8.6614	4.255	5.551	10.93
MultiResGC	16.2	0.903	1.324	4.822	0.4524	0.8423	3.3219	4.3421	10.527	10.725	6.4611	11.814	17.022
OverSegmBP	19.1	1.6918	1.9716	8.4727	0.5126	0.6812	4.6928	3.1915	8.8117	8.8921	6.7420	11.915	15.814

Note: “nonocc” denotes no occlusions; “all” denotes the whole image; “disc” denotes near discontinuities

6.3 Summary

This chapter is the continuation of Chapter 5. It describes the remaining steps of segment-based stereo matching algorithm using cooperative optimization, namely, disparity plane estimation and cooperative optimization of energy function to obtain the refined parameters of the disparity plane. In the disparity plane estimation step, we employ a simple and robust plane fitting method using SVD to solve the least square equation. Due to the sensitivity of the least square function for outliers, we formulate three rules to filter outliers, such as, Cross-checking, Judging reliable segment, and Measuring the distance between the previously determined initial disparity to the current computed disparity. In order to find the similarity of the segmented plane, we design the similarity function. The function is used to merge neighboring disparity planes. The planes properties, such as parallelism and intersection, are proposed as similarity measurements. The similarity function is applied to solve the outliers, occlusion and texture-less problems which improve the accuracy of the disparity map. For the cooperative optimization step, we formulate an energy function for optimization. This energy function considers likelihood, smoothness, and penalty of occlusion. The minimization of this energy function will provide us better disparity plane parameters. Finally, we obtain an accurate disparity map. In the next Chapter, we will introduce the multi-view stereo matching algorithm and depth recovery.

Chapter 7 Multi-view stereo matching and depth recovery

In this chapter, the algorithms of multi-view stereo matching and depth recovery are presented. Firstly, two methods are proposed to solve multi-view stereo correspondence and to obtain the disparity map. They include a local method and a global method, based largely on the proposed algorithm of stereo matching with two views in Chapters 5 and 6. Subsequently, the recovery of the depth of a scene or object after the disparity map is acquired is introduced.

7.1 Multiple views stereo matching

The reconstruction of three-dimensional shape from two or more images is one of the classic research problems in computer vision. The process of acquiring precise 3-D information of an object or scene becomes complicated due to ambiguous local appearances of image pixels, image noise, occlusion, and insufficient texture details. Thus, algorithms that produce a smooth disparity map tend to weaken the details, and those that can extract a detailed map tend to be coarse. To reduce the matching ambiguity, three or more images, which provide more information, are used for stereo matching. Okutomi and Kanade [119] proposed a multi-baseline stereo algorithm, which uses the sum of sums of absolute differences (SSSD) function to overcome the ambiguity. Ueshiba [120] implemented the technique of bidirectional matching for trinocular stereo vision. Li and Jia [121] validated and evaluated the classical binocular algorithm (cooperative algorithm) on multiple-camera stereo system to extract smooth and accurate dense disparity and handle occlusion.

In this thesis, an improved approach to solve multiple views stereo matching with different baselines is proposed. Figure 7.1 shows n stereo pairs with different baselines. These stereo pairs are all assumed to have been rectified, thus, their epipolar lines are all collinear and horizontal. In a stereo correspondence problem, we compute the disparity d , which is the

difference between the corresponding points on the captured images of the same scene. The disparity d is related to the distance Z by

$$d = \frac{\lambda f}{Z} \quad (7.1)$$

where λ and f are the length of the baseline and the focal length, respectively.

Unlike in the case of stereovision with only two images whereby the disparity is evaluated simply between the two images, for a multi-view stereo correspondence problem, merging disparities between reference image (anyone of the selected images) and target images is an important step to find the multi-view stereo correspondence. Two methods are proposed to obtain the disparity map: local method and global method. These two methods are derived based on proposed method in Section 5.2 and Section 6.2.

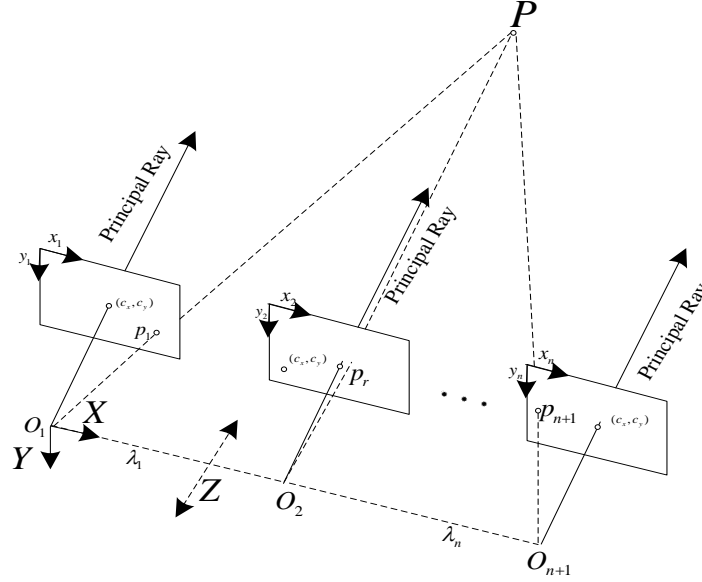


Figure 7.1 Collinear multiple stereo

7.1.1 Applying the local method to obtain multi-view stereo disparity

The biologically and psychophysically inspired adaptive support weights algorithm to obtain the disparity map has been discussed in Chapter 5. The cost function is

$$C(x, y, d) = \frac{\sum \omega_{tot,l} \cdot \omega_{tot,r} \cdot S_{AD}(x, y, d)}{\sum \omega_{tot,l} \cdot \omega_{tot,r}} \quad (7.2)$$

This cost function is primarily designed for the matching of two views in binocular stereovision, thus additional requirement of merging all disparities sets is necessary in the case of multi-views.

For multiple baseline stereo algorithm, given n stereo pairs, the disparity d_i is given by different baseline λ_i (Eq. (7.3)). A comprehensive description of the stereo algorithm is presented using multiple-baseline stereo pairs.

$$d_i = \frac{\lambda_i f}{Z} \quad (7.3)$$

where λ_i is the baseline of the reference image optical center to the i^{th} target image optical center. Z is the depth of scene.

Let

$$\zeta = \frac{1}{Z}$$

Thus, the cost function Eq.(7.2) is modified to become:

$$C(x, y, \zeta) = \frac{\sum \omega_{tot,l} \cdot \omega_{tot,r} \cdot S_{AD}(x, y, \lambda_i f \zeta)}{\sum \omega_{tot,l} \cdot \omega_{tot,r}} \quad (7.4)$$

The objective is to choose proper value of ζ which minimizes Eq. (7.4).

For n stereo pairs with different baselines, the sum of cost function is given as

$$Sum_C(x, y, \zeta) = \sum_{i=1}^n \left(\frac{\sum \omega_{tot,l} \cdot \omega_{tot,r} \cdot S_{AD}(x, y, \lambda_i f \zeta)}{\sum \omega_{tot,l} \cdot \omega_{tot,r}} \right) \quad (7.5)$$

Instead of directly using the inverse distance ζ , normalization of the disparity values of individual stereo pairs by the corresponding values for the largest baseline is performed for the optimization of the algorithm. In stereovision, a short baseline implies the estimated distance would be less precise due to narrow triangulation. For more precise distance estimation, a longer baseline is desired [119].

Suppose $\lambda_1 < \lambda_2 < \dots < \lambda_n$ (n stereo pairs with different baselines in Figure 7.1). We denote the baseline ratio δ , such that

$$\delta_i = \frac{\lambda_i}{\lambda_n} \quad (7.6)$$

Then,

$$\lambda_i f \zeta = \delta_i \lambda_n f \zeta = \delta_i d_{(n)} \quad (7.7)$$

where $d_{(n)}$ is the disparity set for the stereo pair with baseline λ_n .

Substituting this into equation $Sum_C(x, y, \zeta)$

$$Sum_C(x, y, d_{(n)}) = \sum_{i=1}^n \left(\frac{\sum \omega_{tot,l} \cdot \omega_{tot,r} \cdot S_{AD}(x, y, \delta_i d_{(n)})}{\sum \omega_{tot,l} \cdot \omega_{tot,r}} \right) \quad (7.8)$$

The $Sum_C(x, y, d_{(n)})$ function is computed for a range of disparity values at the pixel level and the disparity which gives the minimum value is identified. The determination of disparity based on this approach is therefore easier as this is a local approach.

7.1.2 Applying the global method to obtain multi-view disparity map

The basic principle of global method to obtain multi-view disparity is similar to the algorithm of stereo matching for two views introduced in Chapter 6. The significant steps of the global method, namely, initial disparity map acquisition, disparity plane estimation and cooperative optimization of energy function, are introduced in the following subsections.

1) Initial disparity map acquisition

Figure 7.2 shows the multi-view stereo pairs setup, in which all the images are rectified. There are n target images and one reference image as shown in Figure 7.2 and it is assumed that these image pairs have the same baseline. The biologically and psychophysically inspired adaptive support weights algorithm has been employed to obtain the disparity map. The expression used is given below:

$$C(x, y, d^{(i)}) = \frac{\sum \omega_{tot,l} \cdot \omega_{tot,r} \cdot S_{AD}(x, y, d^{(i)})}{\sum \omega_{tot,l} \cdot \omega_{tot,r}} \quad (7.9)$$

where $d^{(1)}, d^{(2)}, \dots, d^{(n)}$ denote the disparity sets by computing the cost function of reference image and target image 1, reference image and target image 2, \dots , reference image and target image n , respectively. Note that if there are m segments in the reference image, then in each disparity set there will be m disparity planes.

After minimizing the cost function, a series of disparity sets $(d^{(1)}, d^{(2)}, \dots, d^{(n)})$ is obtained.

The next step is to merge the all the disparity sets (n sets).

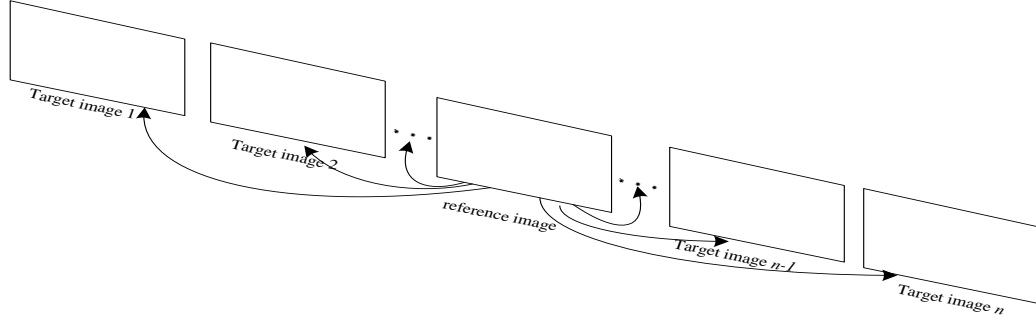


Figure 7.2 The multi-view stereo pairs

2) Disparity plane estimation for multi-view

Before we begin, we would like to re-define the representation for disparity set $d^{(i)}$. There are n disparity sets as mentioned in the previous section. Now, if there are m segments in the reference image, then in each disparity set, there will be m disparity planes. We shall use the notation $d^{(i,j)}$, for $i=1$ to n , and $j = 1$ to m , which means the disparity plane of j^{th} segment in the i^{th} disparity set.

Modeling of the disparity plane using the initial disparities in a segment [109] has been discussed in Chapter 6.

The j^{th} segment in the i^{th} disparity set is defined generally as

$$d^{(i,j)}(x,y) = ax + by + c \quad (7.10)$$

For multi-views, the models of the j^{th} segment in each of the n disparity sets are written as:

$$d^{(1,j)}(x,y) = ax + by + c$$

$$d^{(2,j)}(x,y) = ax + by + c$$

$$d^{(i,j)}(x,y) = ax + by + c$$

$$\vdots$$

$$d^{(n,j)}(x,y) = ax + by + c$$

This series of plane equations representing j^{th} segments in all the disparity sets have the same coefficients a , b and c . This is due to the fact that each segment should contain only unique disparity map. Hence, the coefficients of the disparity planes are computed by the above simultaneous equations. Representing the above series of equations in matrix form:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \dots & \dots & \dots \\ x_p & y_p & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \dots & \dots & \dots \\ x_p & y_p & 1 \\ \dots & \dots & \dots \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \dots & \dots & \dots \\ x_p & y_p & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} d_1^{(1,j)} \\ d_2^{(1,j)} \\ \dots \\ d_p^{(1,j)} \\ d_1^{(2,j)} \\ d_2^{(2,j)} \\ \dots \\ d_p^{(2,j)} \\ \dots \\ d_1^{(n,j)} \\ d_2^{(n,j)} \\ \dots \\ d_p^{(n,j)} \end{bmatrix} \quad (7.11)$$

where p is the number of pixels in the j^{th} segment, $d_k^{(i,j)}$ is the disparity of k^{th} pixel in the j^{th} segment of the i^{th} disparity set ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$ and $k = 1, 2, \dots, p$).

Let

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \dots & \dots & \dots \\ x_p & y_p & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \dots & \dots & \dots \\ x_p & y_p & 1 \\ \dots & \dots & \dots \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \dots & \dots & \dots \\ x_p & y_p & 1 \end{bmatrix} \quad B = \begin{bmatrix} d_1^{(1,j)} \\ d_2^{(1,j)} \\ \dots \\ d_p^{(1,j)} \\ d_1^{(2,j)} \\ d_2^{(2,j)} \\ \dots \\ d_p^{(2,j)} \\ \dots \\ d_1^{(n,j)} \\ d_2^{(n,j)} \\ \dots \\ d_p^{(n,j)} \end{bmatrix}$$

Here, we also use Singular Value Decomposition (SVD) for least square solution (as in Section 6.1.1) to solve the matrix equation:

$$[a, b, c]^T = (A^T A)^+ A^T B \quad (7.12)$$

where $(A^T A)^+$ is the pseudoinverse of $A^T A$, $(A^T A)^+$ can be computed using SVD.

Here,

$$A^T A = \begin{bmatrix} n \sum_{i=1}^p x_i^2 & n \sum_{i=1}^p x_i y_i & n \sum_{i=1}^p x_i \\ n \sum_{i=1}^p x_i y_i & n \sum_{i=1}^p y_i^2 & n \sum_{i=1}^p y_i \\ n \sum_{i=1}^p x_i & n \sum_{i=1}^p y_i & n \end{bmatrix} = n \begin{bmatrix} \sum_{i=1}^p x_i^2 & \sum_{i=1}^p x_i y_i & \sum_{i=1}^p x_i \\ \sum_{i=1}^p x_i y_i & \sum_{i=1}^p y_i^2 & \sum_{i=1}^p y_i \\ \sum_{i=1}^p x_i & \sum_{i=1}^p y_i & 1 \end{bmatrix}$$

$$\begin{aligned}
A^T B &= \begin{bmatrix} x_1 & x_2 & \cdots & x_p & x_1 & x_2 & \cdots & x_p & \cdots & x_1 & x_2 & \cdots & x_p \\ y_1 & y_2 & \cdots & y_p & y_1 & y_2 & \cdots & y_p & \cdots & y_1 & y_2 & \cdots & y_p \\ 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & \cdots & 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} d_1^{(1,j)} \\ d_2^{(1,j)} \\ \cdots \\ d_p^{(1,j)} \\ d_1^{(2,j)} \\ d_2^{(2,j)} \\ \cdots \\ d_p^{(2,j)} \\ \cdots \\ d_1^{(n,j)} \\ d_2^{(n,j)} \\ \cdots \\ d_p^{(n,j)} \end{bmatrix} \\
&= \begin{bmatrix} \sum_{i=1}^p x_i \left(\sum_{l=1}^n d_i^{(l,j)} \right) \\ \sum_{i=1}^p y_i \left(\sum_{l=1}^n d_i^{(l,j)} \right) \\ \sum_{i=1}^p \left(\sum_{l=1}^n d_i^{(l,j)} \right) \end{bmatrix}
\end{aligned}$$

The expression of $A^T A$ after SVD is

$$SVD(A^T A) = UDV^T = [u_1, u_2, \dots, u_p] \begin{bmatrix} D_0 & 0 \\ 0 & 0 \end{bmatrix} [v_1, v_2, \dots, v_n]^T$$

and

$$(A^T A)^+ = [v_1, v_2, \dots, v_m] \begin{bmatrix} D_0^{-1} & 0 \\ 0 & 0 \end{bmatrix} [u_1, u_2, \dots, u_n]^T \quad (7.13)$$

where: vector u_i is the eigenvector of $A^T A(A^T A)^T = (A^T A)^2$,

$$D_0 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r), \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0,$$

σ_i^2 are the nonvanishing eigenvalues of $(A^T A)^2$, or

σ_i are the nonvanishing singular values of $A^T A$,

Vector v_i is the eigenvector of $(A^T A)^T A^T A = (A^T A)^2$.

Here we use the property: vector $u_i = v_i$. The complex computation can then be reduced. We have analyzed the advantages of SVD in Section 6.1.1.

We rewrite Eq. (7.12) to obtain

$$\begin{bmatrix} \sum_{i=1}^p x_i^2 & \sum_{i=1}^p x_i y_i & \sum_{i=1}^p x_i \\ \sum_{i=1}^p x_i y_i & \sum_{i=1}^p y_i^2 & \sum_{i=1}^p y_i \\ \sum_{i=1}^p x_i & \sum_{i=1}^p y_i & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \frac{1}{n} \begin{bmatrix} \sum_{i=1}^p x_i (\sum_{l=1}^n d_i^{(l,j)}) \\ \sum_{i=1}^p y_i (\sum_{l=1}^n d_i^{(l,j)}) \\ \sum_{i=1}^p (\sum_{l=1}^n d_i^{(l,j)}) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^p x_i (\frac{\sum_{l=1}^n d_i^{(l,j)}}{n}) \\ \sum_{i=1}^p y_i (\frac{\sum_{l=1}^n d_i^{(l,j)}}{n}) \\ \sum_{i=1}^p (\frac{\sum_{l=1}^n d_i^{(l,j)}}{n}) \end{bmatrix} \quad (7.14)$$

The solution of Eq. (7.14) will give the coefficients of the disparity plane of the j^{th} segment. We can follow the same method to other segments. At the end of this step, we would have obtained the disparity map of the image.

We also consider the outliers when computing the least square solution for the linear system equation (Eq.(7.14)) by SVD. Three rules are formulated to filter outliers, including cross-checking, assessing reliable region, and measuring the distance between the previously determined disparity to the current computed disparity, which have been discussed in Section 6.1.2.

After having obtained the disparity planes of all the segments, the feasibility of merging neighboring disparity planes is evaluated using the similar measurement (criterion) which has already been described and defined in Section 6.1.3. The next step is to obtain better values of the disparity plane parameters through optimization.

3) Cooperative optimization of energy function

An energy function for multi-view is formulated after the disparity plane estimation. This function is composed of three parts: data cost, penalty of occlusion, and smoothness.

Similar to Section 6.2.2 of Chapter 6 the energy function is expressed by:

$$E_j = E_{j_{data}} + E_{j_{occ}} + E_{j_{sm}} \quad (7.15)$$

The data cost, $E_{j_{data}}$, is written as:

$$E_{j_{data}} = \sum_{i=1}^n \left(\sum_{(x,y) \in N_r} c_{data}(x, y, d_{(x,y)}^{(i,j)}) \right) \quad (7.16)$$

where $d_{(x,y)}^{(i,j)}$ is the disparity of pixel (x, y) of the j^{th} segment in reference image and the i^{th} disparity set.

For penalty of occlusion, $E_{j_{occ}}$, is written as:

$$E_{j_{occ}} = \sum_{i=1}^n (\omega_{occ} \cdot N_{occ}^{(i,j)}) \quad (7.17)$$

where $N_{occ}^{(i,j)}$ is the number of occluded pixels in the j^{th} segment of the i^{th} disparity set.

For smooth term, $E_{j_{sm}}$, is written as:

$$E_{j_{sm}} = \sum_{i=1}^n \left(\sum_{(x,y) \in N_r} s_{sm}(x, y, d_{(x,y)}^{(i,j)}) \right) \quad (7.18)$$

The meanings of all the parameter are defined in Section 6.2.2.

From the three above key steps of the proposed method, namely, initial disparity map acquisition, disparity plane estimation and the energy function formulation, we can minimize the energy function by Pollow's method (see Appendix E) and obtain the best disparity map for multi-view setup.

7.2 Depth recovery

The depth of a scene is recovered using a stereo system model if two or multiple images of the same scene are taken from two or more different camera positions. The significance of the matching algorithm is to find the correspondence between the same points from the captured images from a 3D scene which is projected onto cameras [10]. If such correspondence is identified for each pixel in the images, the resultant map is termed as the dense disparity map. A good revision on stereo matching algorithms generating dense depth maps can be found in [122]. In this section, the algorithms of 3D depth recovery from disparities which are obtained from stereo matching algorithm are introduced.

7.2.1 Triangulation to general stereo pairs

For 3D scene reconstruction from a captured stereo pair of that scene, the simplest situation is when both the intrinsic and extrinsic parameters are available [10]. By referring to Figure 7.3, the left camera coordinate frame is chosen to be the world coordinate system. R denotes the rotational transformation matrix of the right camera frame with respect to the left camera frame. T is the translational transformation of the right camera frame relative to the left camera frame. As shown in Figure 7.3, the point P , is projected to a pair of corresponding points p_l and p_r on the two image planes, lying at the intersection of the two rays from O_l through p_l and from O_r through p_r respectively. In other words, the coordinates of point P can be computed by finding the intersection of the two rays. However, the two rays are mostly

two skew lines in 3D space, which may not intersect each other at a physical point. Therefore, the mid-point theorem is employed to estimate the coordinates of point P (see Figure 7.4).

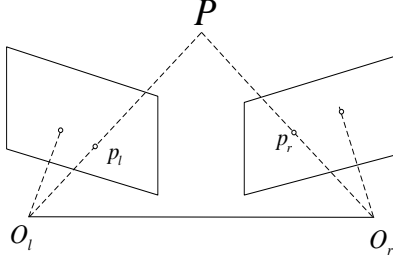


Figure 7.3 Stereo images system

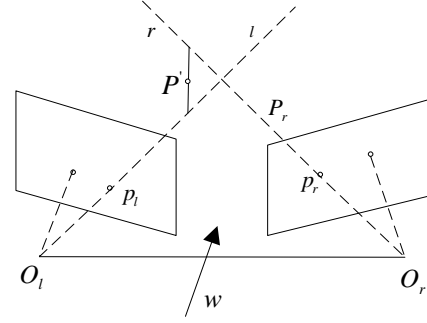


Figure 7.4 Triangulation with nonintersecting

Let $a\mathbf{p}_l$ ($a \in \mathbb{R}$) be the ray, l , through O_l and \mathbf{p}_l . Let $\mathbf{T} + bR^T\mathbf{p}_r$ be the ray, r , through O_r and \mathbf{p}_r , expressed in the left camera frame (which is taken to be the world coordinate frame). Let w be a vector orthogonal to both l and r . Our problem is to determine the midpoint, P' , of the segment parallel to w that joins l and r (see Figure 7.4).

This is simple as the endpoints of the segment, say $a_0\mathbf{p}_l$ and $\mathbf{T} + b_0R^T\mathbf{p}_r$ can be computed by solving the linear system of equations [10]

$$a\mathbf{p}_l - bR^T\mathbf{p}_r + c(\mathbf{p}_l \times R^T\mathbf{p}_r) = \mathbf{T} \quad (7.19)$$

where a_0, b_0 and c_0 are the solution of Eq.(7.19) for obtaining the endpoints of the segment.

7.2.3 Triangulation to rectified stereo pairs

After the rectification of the stereo pairs, the epipolar lines become collinear or parallel to one of the world coordinate axis (which is usually horizontal). Under this condition, Eq. (7.1) can be used to find the 3D depth of the scene or object.

The projection matrices which relates a 3-D point in homogeneous coordinates to a 2D point in homogeneous coordinates is shown as below (Section 3.1) [114]:

$$P_{ppm} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \omega \end{bmatrix} \quad (7.20)$$

where the screen coordinates can be calculated as

$$\left(\frac{x}{\omega}, \frac{y}{\omega} \right),$$

and P_{ppm} is the perspective projection matrix (Section 3.1).

Points in two dimensions can also be reprojected into three dimensions given their screen coordinates and the camera intrinsic matrix. Figure 7.5 shows two parallel cameras, and the two image planes are row-aligned after rectification.

Referring to the Figure 7.5, we have

$$\frac{\lambda - (x_l - x_r)}{Z - f} = \frac{\lambda}{Z} \quad (7.21)$$

$$Z = \frac{\lambda f}{x_l - x_r}$$

where $d = x_l - x_r$, x_l and x_r are the horizontal positions of the points in the left and right image, respectively.

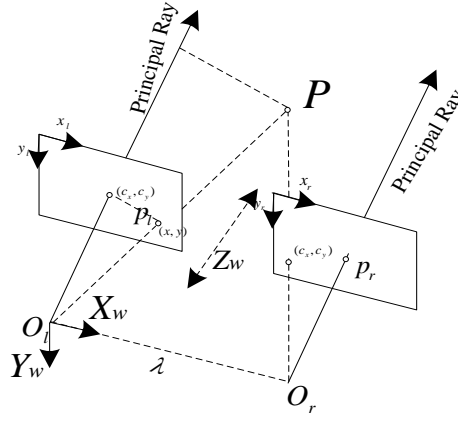


Figure 7.5 Rectified cameras image planes

The coordinates of points in the O_l coordinate system will be referred to as (X, Y, Z) . By referring to Figure 7.5, the similar triangulation of the coordinate of P and p_l can be written as

$$\begin{cases} x - c_x = WX \\ y - c_y = WY \\ f = WZ \end{cases}$$

where W is a scalar, (c_x, c_y) is the center point of the image plane, and (x, y) is the coordinate of the point P projection on left image plane.

Then, for any 3D scene point (X, Y, Z) with disparity d obtained from Eq.(7.21), the equations are

$$\begin{cases} x - c_x = WX \\ y - c_y = WY \\ f = WZ \\ Z = \frac{\lambda f}{d} \end{cases} \quad (7.22)$$

We use matrix Q to express this equation

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \quad (7.23)$$

So, the reprojection matrix is:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & 1/\lambda & 0 \end{bmatrix}$$

Given a two-dimensional homogeneous point and its associated disparity d , projection of the point into three dimensions is carried out using Eq. (7.23) to obtain the 3D coordinates (X, Y, Z) .

From the above development, if we know the disparity and the intrinsic parameters, the reprojection matrix can be easily acquired, and the 3D depth can be recovered directly.

7.3 Experimental results

In this section, we evaluate and validate the algorithms of the multi-views stereo matching and depth recovery.

7.3.1 Multi-view stereo matching algorithm results and discussion

We make use of the standard images (Tsukuba) with the corresponding ground-truth map to carried out the experiment. Figure 7.6 shows the results of our experiments. Figure 7.6 (a), (b) and (c) are Tsukuba images which have been captured by the camera at different positions along the same baseline [14]; Figure 7.6 (d), (e) and (f) are respectively the ground-truth map, result of the local method, and the result of global method.

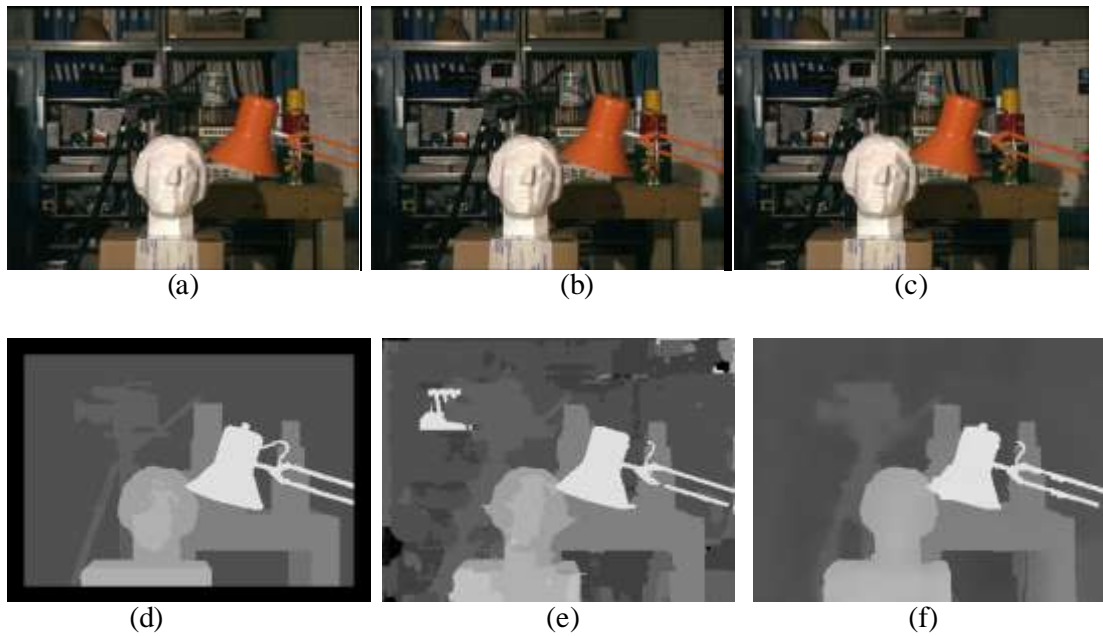


Figure 7.6 Tsukuba images: (a), (b), and (c) are Tsukuba images, (d) ground-truth map, (e) multi-view stereo matching algorithm result (local method), (f) multi-view stereo matching algorithm result (global method)

Table 7.1 shows the comparison of the two-view and multi-view stereo matching algorithm. From the results based on the percentage of bad matching pixels, the results of the global method of the multi-view stereo matching algorithm are better than those of the two-view algorithm as shown in Table 7.1. This is due to the fact that multi-view algorithm provides more information. However, the local method of multi-view stereo matching algorithm does not out-perform the two views stereo matching algorithm. This is due to the fact that the local method is very sensitive to local ambiguities.

Table 7.1 The results of two-view and multi-view stereo matching algorithm

Percentage of bad matching pixels	Reference images	Tsukuba Ground truth
Methods		
the local multi-view stereo matching algorithm		1.345
the global multi-view stereo matching algorithm		1.021
Two views stereo matching algorithm		1.211

To test the multi-view stereo matching algorithms, we use images taken from our single-lens based stereovision system using 4-face prism. The results of rectification have been presented in Chapter 4. Here, Figure 7.7 shows the rectified images. The regions of interest are extracted from the rectified images as shown in squares in Figure 7.7. Figure 7.8 shows the results of the image disparity, which is obtained from the implementation of the global multi-view stereo matching algorithm.

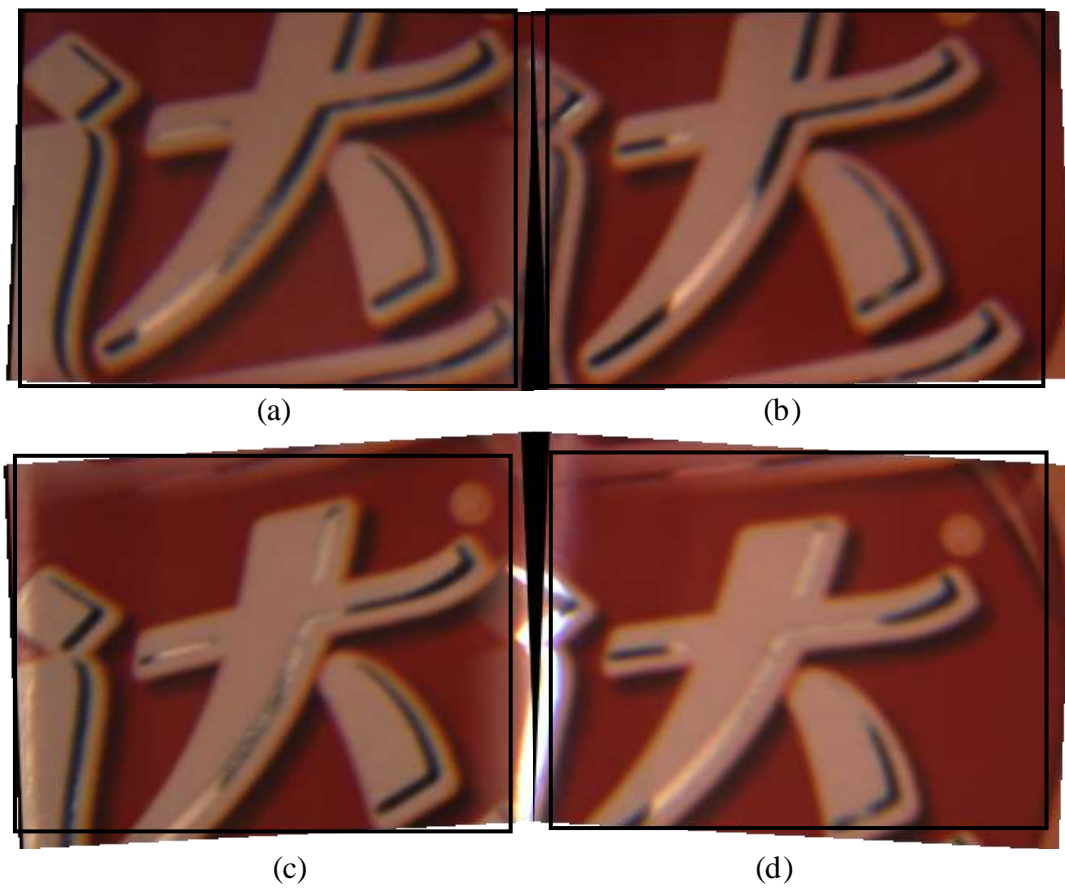


Figure 7.7 The rectified “da” images (a), (b), (c) and (d) by the geometric approach. The unrectified images have been captured by Single-lens stereovision system with 4-face prism



Figure 7.8 “da” images disparity map

7.3.2 Depth recovery results and discussion

In this section, we present some results of depth recovery. We employ the algorithm of triangulation for collinear stereo pairs. Figures 7.9, 7.10 and 7.11 show the results of depth recovery. Figure 7.12 shows the multi-view depth recovery result, the image disparities are obtained using the global method based multi-view stereo matching algorithm.

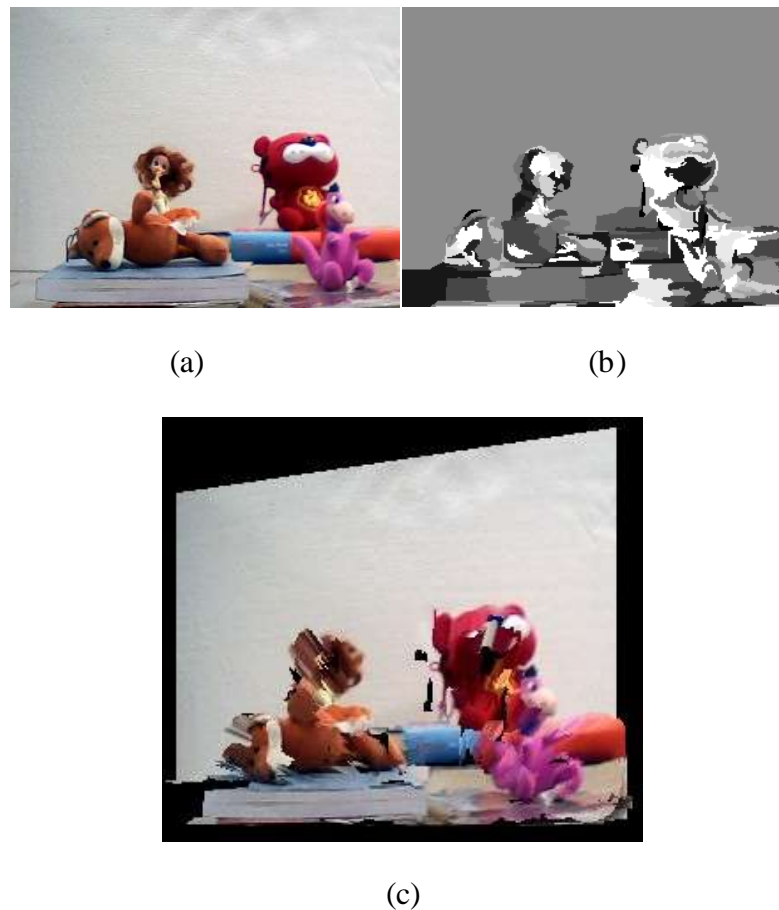


Figure 7.9 “Pet” image depth recovery: (a) original image of pet, (b) the disparity map, and (c) depth reconstruction



Figure 7.10 “Fan” image depth recovery: (a) original image of pan, (b) the disparity map, and (c) depth recovery

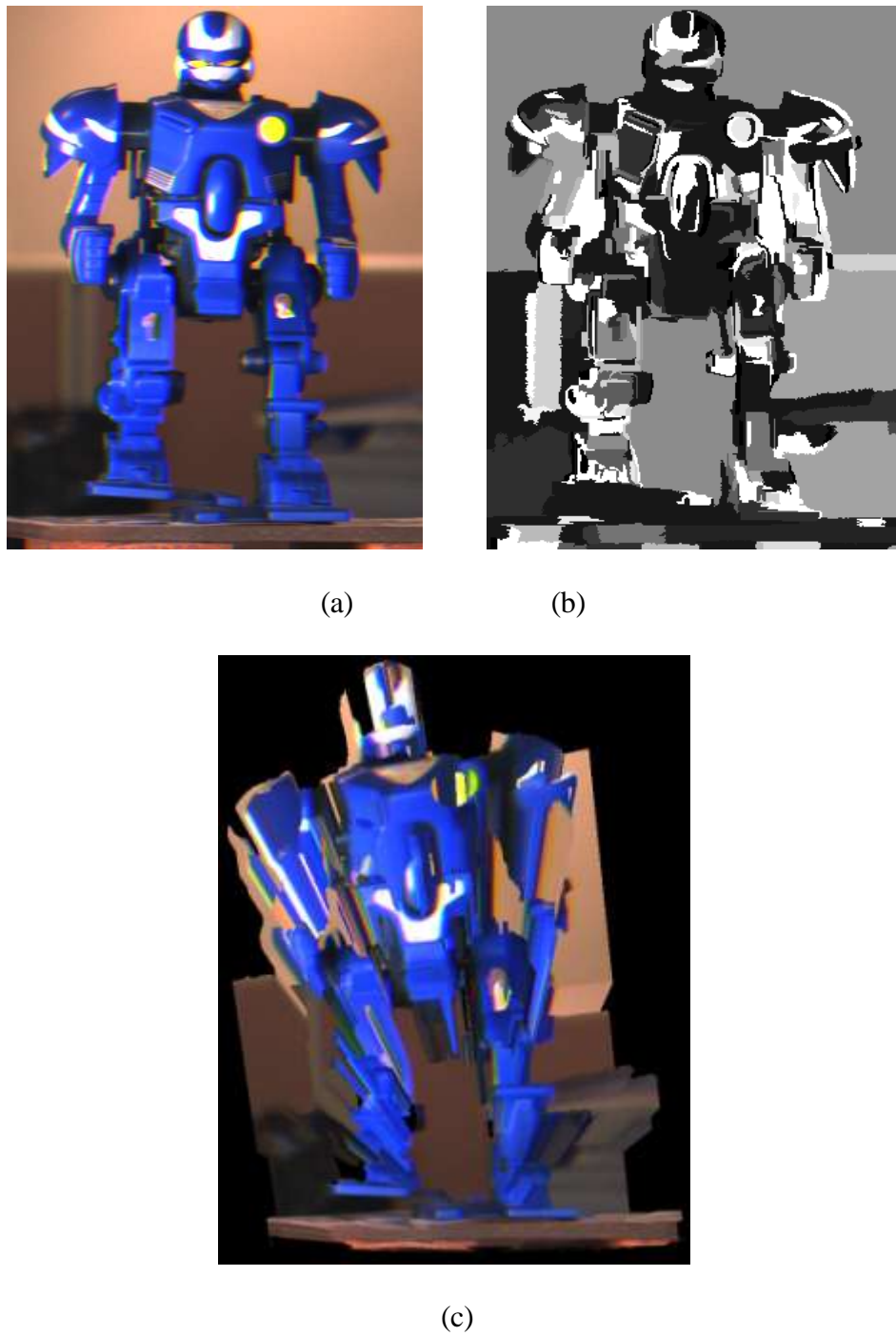


Figure 7.11 “Robot” image depth recovery: (a) original image of robot, (b) the disparity map, and (c) depth recovery

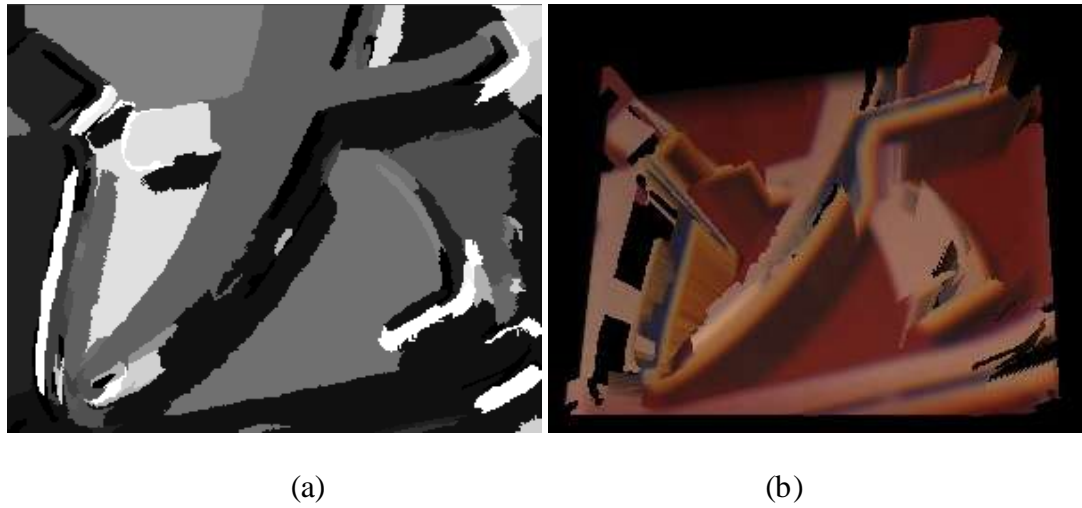


Figure 7.12 “da” image depth recovery: (a) the disparity map of “da”, and (b) depth recovery

We pick some random points on the “robot” image which are shown in Figure 7.13 to test the proposed method to recover the depth of a scene. Table 7.2 shows the recovered depth of the binocular stereovision where the 1st column is the actual depth measured using sensor, the 2nd column is the pixel coordinate in the image, the 3rd column shows the depth recovered using Lim and xiao’s approach [21], the 4th column is the absolute error (1) in percentage which is computed between the values in column 1 and column 3, the 5th column shows the depth recovered using the disparity map based approach, and the 6th column is the absolute error (2) in percentage which is computed between values in the column 1 and column 5. The results of the average absolute error (1) and (2) are shown in the last row. It can be seen from the results that for the depth ranged from 1.504m to 1.523m, the computed disparity map based approach gives an absolute depth recovery error of 0.4839% on average, while the calibration based approach gives an error of 0.62025% under the same condition. The depth recovery approach (computed disparity map based approach) presented in Section 7.2.2 appears to be effective and accurate in determining the depth recovery based on this system.

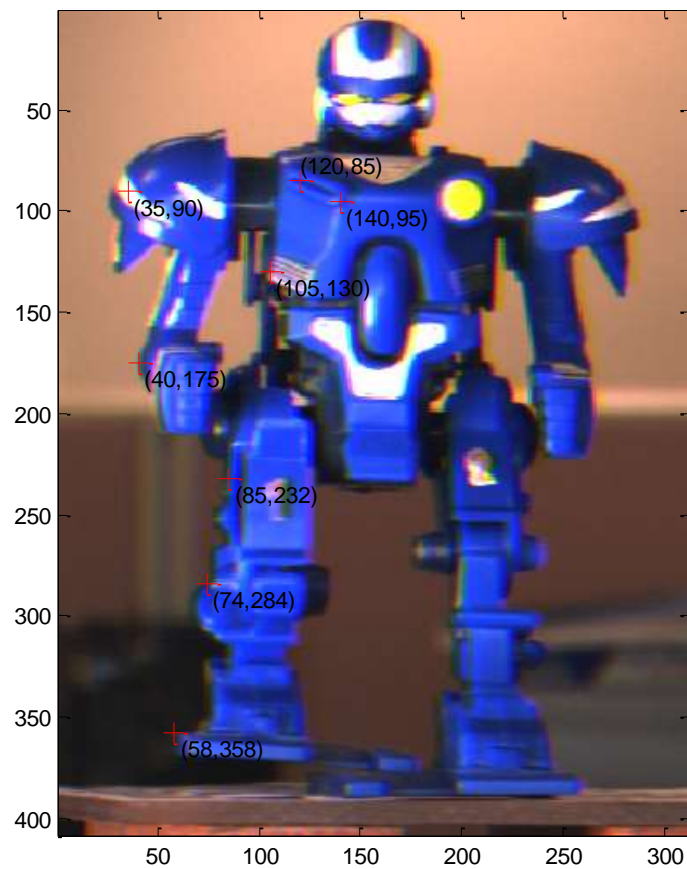


Figure 7.13 Several test points are selected in robot image

Table 7.2 Recovered depth using binocular stereovision

Actual Depth (mm)	Pixel's coordinate in the image	Recovered Depth (mm, Lim and xiao's Approach)	Absolute Error (1) in Percentage (%)	Recovered Depth (mm, Based computed disparity map)	Absolute Error (2) in Percentage (%)
1504	(140,95)	1493.7	0.73	1511.2	0.479
1507	(120,85)	1498.4	0.57	1513.6	0.438
1509	(105,130)	1501.6	0.557	1515.3	0.4175
1512	(85,232)	1502.8	0.661	1519.1	0.463
1514	(35,90)	1505.1	0.588	1521.4	0.4888
1518	(40,175)	1508.2	0.646	1526.2	0.54
1520	(74,285)	1510.4	0.632	1528.2	0.539
1523	(58,358)	1514.2	0.578	1530.7	0.506
AVG			0.62025		0.4839

7.4 Summary

In this chapter, two methods to solve the multi-view stereo correspondence problem are proposed, which include the local method and the global method. The local method, which is the biologically and psychophysically inspired adaptive support weights algorithm (Section 5.2) is applied to determine the disparity map of the multi-view stereo images with different baselines. On the other hand, the global method, which considers the three significant steps, namely, initial disparity map acquisition, disparity plane estimation and cooperative optimization of energy function, is used to determine a more accurate disparity map using optimization of the energy function. The results of multi-view stereo matching algorithm are shown in Section 7.3.1. Comparing the disparity map extracted from the two-view (binocular system) and multi-view (multi-ocular system), the latter gives better results as it provides more information. This also verifies the effectiveness of multi-view algorithm. We also compare the disparity map extracted from multi-view using the local method and the global method, the result of global method is again better than the local method. Next, we introduce the triangulation algorithm to recover the depth of a scene, which is applied to general stereo image pair and rectified stereo image pair. In the rectified stereo image pair, the computed disparity map is used to recover the depth. The results of depth recovery are shown in Section 7.3.2. Comparing to the results of depth recovery using calibration-based approach with our proposed method (Section 7.2.2) shown in Table 7.2, our proposed method is able to obtain more accurate depth of the scene. This verifies the effectiveness of our proposed algorithm.

Chapter 8 Conclusions and future works

The main objective of the work reported in this thesis is to study the recovery of the depth of a 3D scene using a single-lens prism based stereovision system. The work in the area of stereovision involves the following steps: (1) system setup, (2) image acquisition, (3) stereo matching to search for correspondence, (4) determining the disparity map, and (5) depth reconstruction or depth recovery. In this thesis, we deal with the last three steps.

In our single-lens stereovision system, a prism is placed in front of the camera to generate multiple views of the same scene. The number of views depends on the number of faces of the prism. These multi-views (n , *says*) are assumed to have been captured by n number of *virtual cameras*. The advantages of this kind of single-lens stereovision system are its compactness, lower cost, automatic synchronization in view capturing and ease of implementation. Unlike a conventional stereovision system which employs more than one camera, the system used in this thesis employs only one camera. However, having to deal with virtual cameras would add certain degree of difficulty in the analysis of the system.

8.1 Summary and contributions of the thesis

The summary and contributions of this thesis are presented in the following sections:

(1) Stereo vision rectification algorithm

Stereo matching to search for correspondence points is a fastidious task, more so if the image planes are not co-planar. This task can however be facilitated if we can find ways to transform the two image planes to become-planar. This process is known as rectification. In chapter 3 and chapter 4, we propose a geometry based approach for image rectification on un-calibrated single-lens prism based stereovision system [21]. This approach also helps to determine the intrinsic and extrinsic parameters of the system without having to carry out the complex

calibration process. This “soft” approach is also more suitable to analyze a system with cameras which do not exist in reality. Some experiments, designed to test the rectification method have been designed and carried out. The results are encouraging which therefore; validate our geometrical approach in solving rectification problem. Our approach has been extended to model the single-lens based stereovision system using multi-face prism with success.

The success of the rectification process has transformed the multi-view stereovision images to become co-planar, and hence the correspondence search is reduced to a one dimensional problem.

(2) Stereo matching algorithm

Stereo matching is a process to search for corresponding points in all the stereo images in order to determine the disparity map. In Chapters 5, and 6, an algorithm which solves the binocular stereovision correspondence problem for two views is proposed and verified. In these chapters, the algorithm of segment-based stereo matching using cooperative optimization is developed to extract the disparities information from the stereo image pairs. In order to ensure a reliable pixel sets for the segment, outliers are filtered by three rules defined in this work. Lastly, to consolidate the neighboring segments, an improved clustering algorithm is applied to merge them based on the geometrical relationships, such as parallelism and intersection. Subsequently, a new energy function is formulated using cooperative optimization for the refinement of the disparity map in order to extract the final disparity map. Results from the experiments carried out using the Middlebury standard tests and datasets demonstrate that our approach is correct and effective.

In Chapter 7, the multi-view stereo matching algorithm is developed based on the algorithm developed in the two previous chapters for binocular stereovision. There are local and global

methods. In this case, the logic is really to design a way to integrate all the disparity sets between the reference and multi-views images into one final disparity map. Experimental results are also presented to show the effectiveness of the developed algorithm.

(3) Depth recovery algorithm

In chapter 7, an algorithm based triangulation method is developed to recover the depth of scene. This algorithm basically has two forms, one for non-coplanar image planes and the other one for rectified images. The disparity map obtained in the stereo matching step is utilized to recover the depth. The experimental study has validated the developed algorithm.

In brief, the main contributions of this thesis are the algorithms and methods developed in 3-D depth recovery with a single-lens prism based stereovision system. The three following aspects are high-lighted.

- The geometrical approach is proposed to solve the rectification problem of single-lens prism based stereovision system.
- The segment-based stereo matching using cooperative optimization method is proposed to solve the stereo correspondence problem. The method can handle occlusion and, texture-less objects.
- Algorithm to handle the multi-view stereo correspondence problem is also proposed. The underlying principle of the technique is to integrate the disparity sets with well defined rules to obtain the final disparity map. The 3-D depth recovery is done based on the information contained in the final disparity map.

The proposed single-lens prism based stereovision system and its associated 3-D depth recovery approaches derived in this work could have compelling advantages and provide significant improvement in the potential applications, such as indoor robot navigation / object

detection, small size hand-hold stereovision system for dynamic scene, and economic 3D feature checker in industries, etc.

8.2 Limitations and Future works

In this thesis, the developed algorithms for 3-D depth recovery using a single-lens prism based stereovision have been shown to be better than many existing ones. However, there are still some limitations, some of which are given below:

- For the rectification algorithm to work, the optical lens should not have any distortion, and that all the components must be accurately positioned and aligned.
- The single-lens prism based stereovision system has two disadvantages: firstly, all the images captured fall on one CCD matrix of the real camera. This will reduce the resolution of the individual images. Secondly, the common view zone and baseline between the virtual cameras are constrained by the prism size and shape, and hence this system is only useful in close-range stereovision.
- For the stereo matching algorithm, refinement of the designed rules to filter outliers and further optimization of the energy function are needed before better results can be expected.

For future endeavours in this research, some recommendations are suggested as follows:

1. Improve the single-lens based stereovision system. The acquired images using tri-prism are not of the same size. The hardware setup of the system warrants further refinements.
2. Develop rectification algorithm by taking into consideration the lens-distortion and system alignments errors. The geometry-based algorithm does not take into account the lens-distortion problem. It should also be noted that the assumptions that the optical axis of the

camera pass through the apex of the prism and that the back plane of the prism was parallel with the camera image plane would necessarily be true. The contributions of these possible sources or errors should be carefully studied in detail.

3. Reconstruction of a 3-D scene object is one of the main aims in stereovision. In this thesis, some basic ideas to recover the depth of scene from disparity map are introduced. In future work, disparity plane determination and refinement must be examined closely, as they are the important preludes to the determination of accurate disparity map. An accurate reconstruction depends largely on an accurate disparity map.

Bibliography

- [1] R. Szeliski, *Computer Vision: Algorithms and Applications*, Microsoft research, Springer, 2008.
- [2] D. Roble, Vision in Film and Special Effects, *Computer Graphics*, 33(4), pp.58–60, 1999.
- [3] Y.-Y. Chuang, et al., Video Matting of Complex Scenes, *ACM Transactions on Graphics*, 21(3), pp.243–248, 2002.
- [4] N. Snavely, S. M. Seitz, and R. Szeliski, Photo Tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics*, 25(3), pp.835–846, 2006.
- [5] M. Goesele, et al., Multi-view Stereo for Community Photo Collections. In *Tenth International Conference on Computer Vision*, Rio de Janeiro, Brasil, 2007.
- [6] H. Sidenbladh, and M. J. Black, Learning the Statistics of People in Images and Video, *International Journal of Computer Vision*, 54(1), pp.189–209, 2003.
- [7] J. Sivic, C. L. Zitnick, and R. Szeliski, Finding People in Repeated Shots of the Same Scene, In *British Machine Vision Conference (BMVC 2006)*, pp.909–918, Edinburgh, 2006.
- [8] D. Marr, *Vision - A Computational Investigation into the Human Representation and Processing of Visual Information*, Freeman, San Francisco, 1982.
- [9] S. T. Barnard and M. A. Fischler, Computational Stereo, *ACM Computing Surveys*, Vol. 14, pp.553-572, 1982.
- [10] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 2006.
- [11] N. Ayache, F. Lustman, Trinocular Stereo Vision for Robotics, in *IEEE Trans Pattern Anal Mach Intell* 13:73-85, 1991.
- [12] R. Hartley, R. Gupta, Computing Matched-epipolar Projections, In: *CVPR 93*, New York, NJ, pp. 549-555, 1993.

-
- [13] C. Loop, Z. Zhang, Computing Rectifying Homographies for Stereo Vision, In: *CVPR99*, Fort Collins, CO, pp I:125-131, 1999.
- [14] D. Scharstein and R. Szeliski, A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms, *Int. Jour. Computer Vision*, 47(1/2/3): 7-42, 2002.
- [15] Y. Xiao and K. B. Lim, A Single-lens Trinocular Stereovision System Using a 3F Filter, in *IEEE Conference on Robotics, Automation and Mechatronics*, Vol. 1, pp.396-400, 2004.
- [16] Y. Nishimoto, Y. Shirai, A Feature-based Stereo Model Using Small Disparities, in: *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 192-196, 1987.
- [17] W. Teoh and X. D. Zhang, An Inexpensive Stereoscopic Vision System for Robots, *Proceedings of International Conference on Robotics*, pp.186-189, 1984.
- [18] Alexandre R.J. Francois, Gerard G. Medioni, Roman Waupotitsch, Mirror symmetry: 2-view Stereo Geometry, *Image and Vision Computing*, Vol. 2, pp. 137-143, 2003.
- [19] A. Gosthasby and W. A. Gruver, Design of A Single-lens Stereo Camera System, *Pattern Recognition*, Vol. 26: 923-936, 1993.
- [20] D. H. Lee and I. S. Kweon, A Novel Stereo Vision System by Biprism, in *IEEE Transactions on Robotics and Automation*, Vol. 16:5288-541, 2000.
- [21] K. B. Lim and Y. Xiao, Virtual Stereovision System: New Understanding on Single-lens Stereovision Using a Biprism, *Journal of Electronic Imaging*, Vol. 14 (4): 043020-1-043020-11, 2005.
- [22] Y. Xiao and K. B. Lim, A Prism-based Single-lens Stereovision System - from Trinocular to Multi-ocular, *Image and Vision Computing*, 25, 1725-1736, 2007.
- [23] S. Ganapathy, Decomposition of Transformation Matrices for Robot Vision, *Proc. IEEE International Conference on Robotics and Automation*, pp.74-79, 1984.
- [24] O. D. Faugeras and G. Toscani, Calibration Problem for Stereo, *Proceedings of International Conference on Computer Vision & Pattern Recognition*, pp.15-20, 1986.
-

- [25] K. W. Wong, Mathematical Formulation and Digital Analysis in Close Range Photogrammetry, *Photogrammetric Engineering and Remote Sensing*, Vol. 41, pp.1355-1373, 1975.
- [26] I. W. Faig, Calibration of Close Range Photogrammetric Systems: Mathematical Formulation, *Photogrammetric Eng. Remote Sensing*, Vol. 41, pp.1479-1486, 1975.
- [27] H. Bacakoglu and M. S. Kamel, A Three-Step Camera Calibration Method, In *IEEE Transactions on Instrumentation and Measurement*, Vol. 46, pp.1165-1172, 1997.
- [28] H. Gao, C. Wu, L. Gao and B. Li, An Improved Two-Stage Camera Calibration Method, *Proceedings of the 6th World Congress on Intelligent Control and Automation*, pp.9514-9518, 2006.
- [29] Z. Zhang, A Flexible New Technique for Camera Calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.22, pp.1330-1334, 2000.
- [30] Z. Zhang, Determining the Epipolar Geometry and Its Uncertainty: A Review, *International Journal of Computer Vision*, Vol. 27, pp.161-195, 1998.
- [31] Lamdan and Wolfson, The Trifocal Tensor,<http://www.robots.ox.ac.uk/~vgg/hzbook/hzbook2/HZtrifocal.pdf>.
- [32] L. F. Cheong, Lecture- 3D Computer Vision, <http://courses.nus.edu.sg/course/electf/ee6901>.
- [33] R. Hartley et al, Stereo from Uncalibrated Cameras, *Computer Vision and Pattern Recognition*, pp.761-764, 1992.
- [34] Dhond UR, Aggarwal JK, Structure from Stereo-A Review, *IEEE Trans Syst Man Cybern*,19(6): 1489-1510, 1989.
- [35] O. Faugeras, Three-Dimensional Computer Vision: A Geometric Viewpoint, *the MIT Press*, Cambridge, Mass, 1993.
- [36] C. C. Slama, editor, Manual of Photogrammetry, *American Society of Photogrammetry*, Falls Church, Va, fourth edition, 1980.

- [37] R. Hartley, Theory and Practice of Projective Rectification, *International Journal of Computer Vision*, 35(2): 1-16, 1999.
- [38] A. Fusiello, E. Trucco, A. Verri, Rectification with Unconstrained Stereo Geometry, *Research Memorandum RM/98/12*, CEE Dept., Heriot-Watt University, Edinburgh, UK. <ftp://ftp.sci.univr.it/pub/Papers/Fusiello/RM-98-12.ps.gz>, 1998.
- [39] K. A. Al-Shalfan, J. G. B. Haigh, S. S. Ipson, Direct Algorithm for Rectifying Pairs of Uncalibrated Images, *Electronics Letters* 36 (5) 419-420. March, 2000.
- [40] F. Isgro, E. Trucco, Projective Rectification without Epipolar Geometry. In: *CVPR99*, Fort Collins, CO, pp. I:125-131, 1999.
- [41] Hsien-Huang P. Wu, Yu-Hua Yu, Projective Rectification with Reduced Geometric Distortion for Stereo Vision and Stereoscopic Video, *Journal of Intelligent and Robotic Systems*, 42(1): Pages 71-94, Jan 2005.
- [42] R. Hartley, P. Sturm, Triangulation, *Compute Vision Image Understanding* 68(2): 146-157, 1997.
- [43] B. Caprile, V. Torre, Using Vanishing Points for Camera Calibration, *International Journal of Computer Vision* 4: 127-140, 1990.
- [44] A. Fusiello, E. Trucco, A. Verri, A. compact, Algorithm for Rectification of Stereo Pairs, *Machine Vision Applications* 12 (1) : 16-22, 2000.
- [45] N. Ayache, F. Lustman, Trinocular Stereo Vision for Robotics, *IEEE Trans Pattern Anal Mach Intell* 13:73-85, 1991.
- [46] Papadimitriou DV, Dennis TJ, Epipolar Line Estimation and Rectification for Stereo Images Pairs, *IEEE Trans Image Process* 3(4): 672-676, 1996.
- [47] L. Robert, C. Zeller, O. Faugeras, M. Hebert, Applications of Nonmetric Vision to Some Visually Guided Robotics Tasks. In: *Aloimonos Y(ed) Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, Chap.5. Lawrence Erlbaum Assoc., pp. 89-134, 1997.

-
- [48] L. Robert, Camera Calibration without Feature Extraction, *Compute vision, Graphics Image Process* 63(2): 314-325, 1996.
- [49] N. Ayache, C. Hansen, Rectification of Images for Binocular and Trinocular Stereovision, In *Proceedings of International Conference on Pattern Recognition*, Ergif Palace Hotel, Rome, Italy vol. 1, 11-16. November, 1988.
- [50] J. Shao, C. Fraser, Rectification and Matching of Trinocular Imagery, *Geomatics Research Australasia* 71, 73-86, 1999.
- [51] <http://www.ptgrey.com>.
- [52] R. T. Collins, A Space-sweep Approach to True Multi-image Matching, In *CVPR*, page 358-363, 1996.
- [53] A. F. Bobick and S. S. Intille, Large occlusion stereo, *IJCV*, 33(3): 181-200, 1999.
- [54] B. K. P. Horn and B. G. Schunck, Determining Optical Flow, *Artificial Intelligence in Perspective*, pp.81-87, 1994.
- [55] V. Venkateswar and R. Chellappa, Hierarchical Stereo and Motion Correspondence Using Feature Grouping, *International Journal of Computer Vision*, Vol.15, pp.245-269, 1995.
- [56] S. Todorovic and N. Ahuja, Region-based Hierarchical Image Matching, *International Journal of Computer Vision*, Vol.78, pp.47-66, 2007.
- [57] T. Kanade, H. Kano, and S. Kimura, Development of a Video-rate Stereo Machine, in *Image Understanding Workshop*, Monterey, CA, pp. 549-557, 1994.
- [58] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, New York: McGraw-Hill, 1990.
- [59] Y. Ohta and T. Kanade, Stereo by Intra- and Intra-Scanline Search Using Dynamic Programming, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7 pp. 139-154, 1985.

-
- [60] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs, A Maximum Likelihood Stereo Algorithm, *Computer Vision and Image Understanding*, vol.63, pp. 542-567, 1996.
- [61] S.S. Intille and A.F. Bobick, Incorporating Intensity Edges in the Recovery of Occlusion Regions, *Proc. International Conference on Pattern Recognition*, vol. 1, pp. 674-677, 1994.
- [62] H.H. Baker, Depth from Edge and Intensity Based Stereo, Technical Report AIM-347, Artificial Intelligence Laboratory, Stanford Univ., 1982.
- [63] P.N. Belhumeur, A Bayesian Approach to Binocular Stereopsis, *International Journal Computer Vision*, vol.19, no. 3, pp. 237-260, 1996.
- [64] S. Birchfield and C. Tomasi, Depth Discontinuities by Pixel-to-Pixel Stereo, *Proc. IEEE International Conference of Computer Vision*, pp. 1073-1080, 1998.
- [65] H. Zhao, Global Optimal Surface from Stereo, *Proc. International Conference on Pattern Recognition*, vol.1, pp. 101-104, 2000.
- [66] I. Thomos, S. Malasiotis, and M.G. Strintzis, Optimized Block Based Disparity Estimation in Stereo Systems Using a Maximum-Flow Approach, *Proc. SIBGRAPI'98 Conf.*, 1995.
- [67] S. Roy and I.J. Cox, A Maximum-Flow Formulation of the N-Camera Stereo Correspondence Problem, *Proc. International Conference on Computer Vision*, pp.492-499, 1998.
- [68] Y. Boykov, V. Kolmogorov, An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, *Proc. Third International Workshop Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2001.
- [69] V. Kolmogorov and R. Zabih, Computing Visual Correspondence with Occlusions Using Graph Cuts, *Proc. International Conference on Computer Vision*, 2001.
- [70] X. Huang, A Cooperative Search Approach to Global Optimization, *Proceedings of the First International Conference on Optimization Methods and Software*, vol. December, p. 140, Hangzhou, P.R. China, 2002.
-

- [71] X. Huang, Cooperative Optimization for Solving Large Scale Combinatorial Problems, in *Theory and Algorithms for Cooperative Systems*, ser. Series on Computers and Operations Research. World Scientific, pp. 117–156, 2004.
- [72] X. Huang, A General Framework for Constructing Cooperative Global Optimization Algorithms, in *Frontiers in Global Optimization*, ser. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, pp. 179–221, 2003.
- [73] X. Huang, A General Global Optimization Algorithm for Energy Minimization from Stereo Matching, in *ACCV*, Korea, pp. 480–485, 2004.
- [74] P. Dev, Segmentation Processes in Visual Perception: A Cooperative Neural Model, Coins Technical Report 74C-5, University of Massachusetts at Amherst, 1974.
- [75] D. Marr, Vision, *W.H. Freeman and Company*, New York, 1982.
- [76] D. Scharstein and R. Szeliski, Stereo Matching with Nonlinear Diffusion, *IJCV*, 28(2): 155-174, 1998.
- [77] C. Zintnick, T. Kanade, A Cooperative Algorithm for Stereo Matching and Occlusion Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (7), 675-684, 2002.
- [78] Y. Zhang, C. Kambhamettu, Stereo Matching with Segmentation-based Cooperation, *European Conference on Computer Vision*, 556-571, 2003.
- [79] Y. Ruichek, Multilevel and Neural-network-based Stereo Matching Method for Real-time Obstacle Detection Using Linear Cameras, *IEEE Trans on Intelligent Transportation Systems*, 6(1): 54-62, 2005.
- [80] X. J. Hua, M. Yokomichi, M. Kono, Stereo Correspondence Using Color Based on Competitive-cooperative Neural Network, *Proc of the 6th International Conference on Parallel and Distributed Computing, Applications and Technologies*, Denver, pp.856-860, 2005.

- [81] J. Shah, A Nonlinear Diffusion Model for Discontinuous Disparity and Half-Occlusions in Stereo, *Proc. Computer Vision and Pattern Recognition*, pp. 34-40, 1993.
- [82] D. Scharstein and R. Szeliski, Stereo Matching with Non-Linear Diffusion, *International Journal of J. Computer Vision*, vol. 28, no. 2, pp. 155-174, 1998.
- [83] A.-R. Mansouri, A. Mitiche, and J. Konrad, Selective Image Diffusion: Application to Disparity Estimation, *Proc. International Conference on Image Processing*, vol. 3, pp. 284-288, 1998.
- [84] P. Fua and Y. G. Leclerc, Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading, *International Journal of Computer Vision*, vol. 16, pp. 35-56, 1995.
- [85] O. Faugeras and R. Keriven, Variational Principles, Surface Evolution, PDE's, Level Set Methods, and the stereo Problem, *IEEE Trans. Image Processing*, vol. 7, pp. 336-344, 1998.
- [86] K. N. Kutulakos and S. M. Seitz, A Theory of Shape by Space Carving, *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199-218, 2000.
- [87] M. Bleyer and M. Gelautz, A Layered Stereo Matching Algorithm Using Image Segmentation and Global Visibility Constraints, *ISPRS Journal of Photogrammetry and remote sensing*, 59(3): 128-150, May 2005.
- [88] Q. Yang, L. Wang, R. Yang, et al., Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation and Occlusion Handling, In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 31, pp. 492-504, 2009.
- [89] A. Klus, M. Smrmann, and K. Karner, Segment-Based stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure, *ICPR 2006*, Vol.3, pp. 15-18, 2006.
- [90] W. Hoff and N. Ahuja, Surfaces from stereo: Integrating feature matching, disparity estimation and contour detection, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 2, pp. 121-136, 1989.

-
- [91] V. P. Lee, Stereovision Using A Single CCD Camera, Master's thesis, Department of Mechanical Engineering, NUS, 2001.
- [92] Y. Xiao, Stereovision Using Single CCD Camera, Technical Report, Control and Mechantronics Laboratory, Department of Mechanical Engineering, NUS, 2000.
- [93] L. C. Ng, Stereo-vision Using Single CCD Camera, Technical Report, Control and Mechantronics Laboratory, Department of Mechanical Engineering, NUS, 2001.
- [94] C. W. Tan, Stereo-vision Using Single CCD Camera, Technical Report, Control and Mechantronics Laboratory, Department of Mechanical Engineering, NUS, 2003.
- [95] R. Y. Tsai, A Versatile Camera Calibration Technique for high-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses, *IEEE Journal of Robotics and Automation*, Vol. RA-3, pp.323-344, 1987.
- [96] H. Maas, Image Sequence Based Automatic Multi-Camera System Calibration Techniques, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 54, Issue 5-6, pp. 352-359, 1999.
- [97] F. Pedersini, A. Sarti and S. Tubaro, Accurate and Simple Geometric Calibration of Multi-camera Systems, *Signal Processing*, Vol. 77, Issue 3, pp. 309-334, 1999.
- [98] B. D. Olsen and A. Hoover, Calibrating A Camera Network Using A Domino Grid, *Pattern Recognition*, Vol. 34, Issue 5, pp. 1105-1117, 2001.
- [99] D. Comaniciu, and P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603-619, 2002.
- [100] E. M. Riseman, M. A. Arbib, Computational Techniques in the Visual Segmentation of Static scenes, *Computer Vision Graphics Image Process*, 6 221-276, 1977.
- [101] W. Skarbek, A. Koschan, Colour image segmentation: A Survey, Technical Report, Tech. Univ. of Berlin, October 1994.
- [102] A. Koestler, The Ghost in the Machine, Penguin reprint, 1990, ISBN 0140191925.
-

-
- [103] A. Oliver et al., A Review of Automatic Mass Detection and Segmentation in Mammographic Images, *Medical Image Analysis* 14, 87–110, 2010.
- [104] K. S. Fu, J. K. Mui, A Survey on Image Segmentation, *Pattern Recogn.* 13, pp.3–16, 1981.
- [105] A. Jain & F. Farrokhnia, Unsupervised Texture Segmentation Using Gabor Filters, In: *Pattern Recognition*. Vol. 24, Nr. 12, S. 1167–1186, 1991.
- [106] A. K. Jain, M. N. Murthy, Flynn, P.J., Data Clustering: A Review, *ACM: Computing Surveys* 31 (3), 264–323, 1999.
- [107] K. Fukunaga and L. D. Hostetler, The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition, *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.
- [108] M. Tabb and N. Ahuja, Multiscale Image Segmentation by Integrated Edge and Region Detection, *IEEE Trans. Image Process.*, vol. 6, pp. 642–655, 1997.
- [109] H. Tao, H.S. Sawhney, R. Kumar, A Global Matching Framework for Stereo Computation, in: *ICCV*, vol. 1, 2001, pp. 532–539.
- [110] Q. Ke, T. Kanade, A Subspace Approach to Layer Extraction, in: *Conference on Computer Vision and Pattern Recognition*, pp. 255–262, 2001.
- [111] D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice Hall, Upper Saddle River, NJ, USA, 2002.
- [112] G. Kanizsa, Grammatica del Vedere/ La Grammaire du Voir, Il Mulino, Bologna/ Editions Diderot, arts et sciences, 1980/ 1997.
- [113] M. Wertheimer, Untersuchungen Zur lehre Der Gestalt, II. *Psychologische Forschung*, 4: 301–350, 1923.
- [114] J. C. Pinoli, J. Debayle, Logarithmic Adaptive Neighborhood Image Processing (LANIP): Introduction, Connections to Human Brightness Perception, and Application Issues, *EURASIP Journal on Advances in Signal Processing*, pp. (1) 22, 2007
-

-
- [115] K. J. Yoon, I. S. Kweon, Adaptive Support-weight Approach for Correspondence Search, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 (4): 650-656, 2006.
- [116] S. Krishnamachari, M. Abdel-Mottaleb, Hierarchical Clustering Algorithm for Fast Image Retrieval, *part of the IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases VII*, San Jose, California, pp. 427-435, January 1999.
- [117] X. Huang. Cooperative Optimization for Energy Minimization: A Case Study of Stereo Matching, <http://front.math.ucdavis.edu/author/X.Huang,cs.CV/071057>, Jan 2007.
- [118] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, Interactive Image Segmentation using an adaptive GMMRF model, In *Proc. European Conf. Computer Vision*, 2004.
- [119] M. Okutomi and T. Kanade, A Multiple-baseline Stereo, *IEEE Trans. Pattern Anal. Mach. Intell.* 15(4), 353-363, 1993.
- [120] T. Ueshiba, An Efficient Implementation Technique of Bidirectional Matching for Real-time Trinocular Stereo Vision”, in *IEEE International conference on Pattern Recognition*, Vol I, pp. 1076-1079, 2006.
- [121] M. Li and Y. Jia, Trinocular Cooperative Stereo Vision and Occlusion Detection”, in *IEEE International conference on Robotics and Biomimetrics*, pp. 1129-1133, 2006.
- [122] L. Di Stefano, M. Marchionni, S. Mattoccia, A PC-based Real-Time Stereo Vision System Machine, *GRAPHICS & VISION* vol. 13, no.3, 2004, pp. 197-220
- [123] Kah Bin Lim, Daolei Wang, Wei loon Kee, Virtual Cameras Rectification with Geometrical Approach on Single-lens stereovision Using a biprism”, *Journal of Electronic Imaging*, 21(2), 023003, 2012.
- [124] Luping An, Yunde Jia, An Efficient Rectification Method for Trinocular Stereovision, *International Conference on Pattern Recognition (ICPR'04)*, vol. 4, pp.56-59, 2004
- [125] A. Eusiello, E. Trucco, A. Verri, A Compact Algorithm for Rectification of Stereo Pairs”, *Machine Vision and Applications*, 12: 16-22, 2000.

-
- [126] L. Nalpantidis, A. Gasteratos, Biologically and psychophysically inspired adaptive support weights algorithm for stereo correspondence, *Robotics and Autonomous System* 58, pp.457-464, 2010.
- [127] Li Hong, George Chen, Segment-based Stereo Matching Using Graph Cuts, *Computer Vision and Pattern Recognition*, Vol.1, pp.74-81, 2004.

Appendices

Appendix A:

The Mid-point Theorem

Two straight lines in 3D do not intersect and are not parallel to each other have a unique shortest distance, which is probably the case that needs to be handled after getting the expression of line RJ and line NL . Figure 1 illustrates this scenario, in which, two non-parallel and non-intersecting lines AB and CD are shown. The shortest distance between them is assumed to be given by EF .

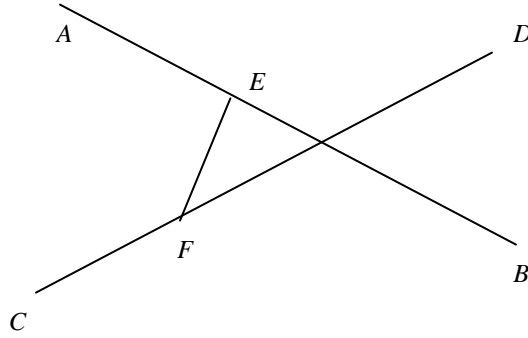


Figure 1 Illustration of the shortest segment connecting two non-intersecting, and non-parallel lines

It is assumed that line AB and line CD do not intersect and are also not parallel lines, and line EF is perpendicular to line AB and line CD and its length is the shortest distance between line AB and line CD . Line AB and line CD are represented by the following expressions:

$$\begin{aligned} P_{AB} &= P_A + K_{AB}(P_B - P_A), \\ P_{CD} &= P_C + K_{CD}(P_D - P_C), \end{aligned} \quad (1)$$

where P_{AB} and P_{CD} are any points on line AB and line CD respectively, and K_{AB} and K_{CD} are corresponding parameters, the value of which depending on the chosen P_{AB} and P_{CD} respectively.

Point E and point F can then be represented as:

$$\begin{aligned} P_E &= P_A + K_{AB}(P_B - P_A), \\ P_F &= P_C + K_{CD}(P_D - P_C). \end{aligned} \quad (2)$$

As line EF is perpendicular to line AB and line CD , the following expression can be obtained:

$$\begin{aligned} (P_E - P_F) \bullet (P_B - P_A) &= 0, \\ (P_E - P_F) \bullet (P_D - P_C) &= 0. \end{aligned} \quad (3)$$

Replacing P_E and P_F in Equation (3) using the Equation (2):

$$\begin{aligned} ((P_A + K_{AB}(P_B - P_A)) - (P_C + K_{CD}(P_D - P_C))) \bullet (P_B - P_A) &= 0, \\ ((P_A + K_{AB}(P_B - P_A)) - (P_C + K_{CD}(P_D - P_C))) \bullet (P_D - P_C) &= 0. \end{aligned}$$

Solving the proceeding equations for the corresponding parameters K_{AB} and K_{CD} for point E on line AB and point F on line CD :

$$\begin{aligned} K_{AB} &= \frac{M_{ACDC}M_{DCBA} - M_{ACBA}M_{DCDC}}{M_{BABA}M_{DCDC} - M_{DCBA}^2}, \\ K_{CD} &= \frac{M_{ACDC} + M_{DCBA} \bullet K_{AB}}{M_{DCDC}}, \end{aligned} \quad (4)$$

where

$$M_{1234} = (x_1 - x_2)(x_3 - x_4) + (y_1 - y_2)(y_3 - y_4) + (z_1 - z_2)(z_3 - z_4).$$

Once the corresponding parameters K_{AB} and K_{CD} for point E and F are found respectively, point E and F can be determined easily and the mid-point of segment EF is taken to be the lens center of virtual camera (i.e. point F').

Appendix B:

The relationship of three views

Refer to the figure 1, three views can be thought of as 3 stereo pairs which are (Cam1, Cam2), (Cam2, Cam3), and (Cam3, Cam1). We can generate some constraint using the epipolar constraint.

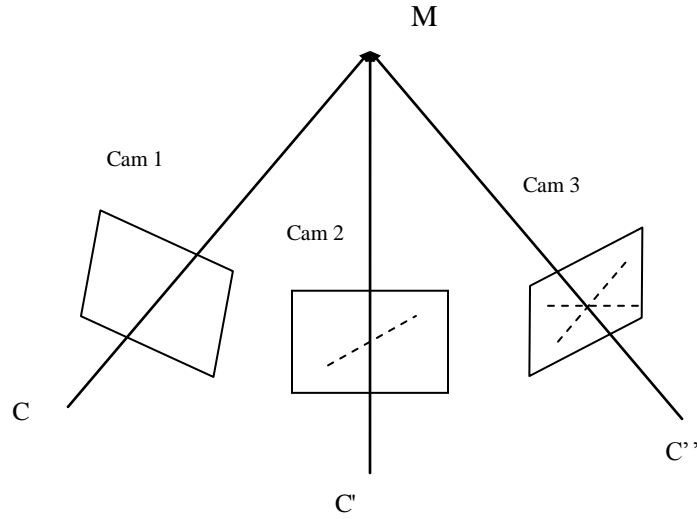


Figure 1 Three views geometry

Assume that we have matched eight points over the three views. Using the eight point algorithm, we can now compute the fundamental matrices F_{12} , F_{23} , F_{31} , respectively, where F_{ij} corresponds to the (Cam i , Cam j) pair.

Next, for any point p in Cam 1's image, we can compute the epipolar line in the Cam 3 image, using the matrix F_{31} . Call it line l_1 . If we have already computed a dense stereo match between the images of Cam 1 and Cam 2, then we would know the location of p in Cam 2 (call it p')

For the point p' in Cam 2, use the fundamental matrix F_{23} to generate the epipolar line in Cam 3. Call it line l_2 . The intersection of l_1 and l_2 gives us the estimation of the location of the corresponding point of p and p' in third image (called it p''). The Moral of the Story: If you have located eight (or more) point matches over three views, and have managed to compute dense point matches between any two of the views, you can “reproject” the dense point matches to the third view using the epipolar line constraint.

When C is in the focal plane of the left camera, the right epipole is at infinity, and the epipolar lines form a bundle of parallel lines in the right image. A very special case is when both epipoles are at infinity, which happens when the line CC' (the baseline) is contained in both focal planes, i.e., the retinal planes are parallel to the baseline. Epipolar lines, then, form a bundle of parallel lines in both images. Any pair of images can be transformed so that the epipolar lines are parallel and horizontal in each image. This procedure is called *rectification*.

Appendix C:**Brief Review of Image Segmentation**

Image segmentation is a critical component in many machine vision and information retrieval systems. It is typically used to partition images into regions that are in some sense of homogeneous, or have some semantic significance, thus providing subsequent processing stages high-level information about the scene structure. To be more exact, segmentation is the division of an image into spatially continuous, disjoint and homogeneous regions. Segmentation is powerful and it has been suggested that image analysis leads to meaningful objects only when the image is segmented in ‘homogenous’ areas [100, 101] or into ‘relatively homogeneous areas’. The latter term reflects the ‘near-decomposability’ of natural systems as laid out by Koestler [102] and we explicitly address a certain remaining internal heterogeneity. The key is that the internal heterogeneity of a parameter under consideration is lower than the heterogeneity compared with its neighboring areas. The diverse requirements of systems that use segmentation have led to the development of segmentation algorithms that vary widely in both algorithmic approach and the quality and nature of the segmentation produced. Some applications simply require the image to be divided into coarse homogeneous regions, others require rich semantic objects. For some applications, precision is paramount, for others speed and automation are more important.

In generic computer vision terminology, segmentation techniques can be divided into unsupervised and supervised approaches [103]. Supervised segmentation or model-based methods rely on the prior knowledge about the object and background regions to be segmented. The prior information is used to determine if the specific regions are present within an image or not.

Alternatively, unsupervised segmentation partitions an image into a set of regions which are distinct and uniform with respect to some specific properties, such as grey-level, texture or colour. Classical approaches to solve unsupervised segmentation are divided in three major groups [104]:

- Region-based methods

Region-based methods divide the image into homogeneous and spatially connected regions. It can be divided into region growing, merging and splitting techniques and their combinations. Many region growing algorithms aggregate pixels starting with a set of seed points. The neighboring pixels are then joined to these initial ‘regions’ and the process is continued until a certain threshold is reached. This threshold is normally a homogeneity criterion or a combination of size and homogeneity.

- Contour-based methods

Contour-based methods rely on the boundaries of the regions. There are various ways to delineate boundaries but in general the first step of any edge based segmentation methods is edge detection which consists of three steps [105]: filtering, enhancement and detection. Filtering step is usually necessary in decreasing the noise in the image. The enhancement aims to reveal of the local changes in intensities. One possibility to implement the enhancement step is high-pass filtering. Finally, the actual edges are detected from the enhanced data using thresholding technique. Finally, the detected edge points have to be linked to form the region boundaries and the regions have to be labeled.

- Clustering methods

Clustering methods which group those pixels which have the same properties might result in non-connected regions. Clustering methods are one of the most commonly used techniques for image segmentation, as discussed in the review by Jain et al.[106], and also for mass detection and/or segmentation. Based on the work of Jain et al., clustering techniques can be divided into hierarchical and partitional algorithms, where the main difference between them is that the hierarchical methods produce a nested series of partitions while partitional methods produce only a single partition. Although hierarchical methods can be more accurate, partitional methods are used in applications involving large datasets, like the ones related to images, as the use of nested partitions is computationally prohibitive. However, partitional algorithms have two main disadvantages: (1) a priori, the number of regions that are in the image has to be known, and (2) clustering algorithms do not use spatial information inherent to the image.

Appendix D:

The detailed definitions of group rules and The Weber-Fechner law

1. The detailed definitions of group rules

Gestalt theory starts with the assumption of active grouping laws in visual perception [Kan97, Wer23] [112, 113]. These groups are identifiable with subsets of the retina. The detail of the gestalt rules by which elements tend to be associated together and interpreted as a group is presented, such as *Vicinity (Proximity)*, *Similarity*, *Continuity*, *Common fate*, *Closure*, *Parallelism*, and *Symmetry*.

(a) *Vicinity (Proximity)*: elements those are close to each other, which apply when distance between elements is small enough with respect to the rest.

(b) *Similarity*: elements similar in an attribute, which leads us to integrate into groups if they are similar to each other.

(c) *Continuity*: the law of continuity holds that points that are connected by straight or curving lines are seen in a way that follows the smoothest path. Rather than seeing separate lines and angles, lines are seen as belonging together.

(d) *Common fate*: elements that exhibit similar behavior.

(e) *Closure*: elements that could provide closed curves. Things are grouped together if they seem to complete some entity.

(f) *Parallelism*: elements that seem to be parallel, which applies to group the two parallel curves, perceived as the boundaries of a constant width object.

(g) *Symmetry*: elements that exhibit a larger symmetry, which applies to group any set of objects which is symmetric with respect to some straight line.

2. The Weber-Fechner law

The mathematical expression of this psychophysical law can be derived considering that the change of perception is proportional to the relative change of the causing stimulus. The mathematical expression of this psychophysical law can be derived considering that the change of perception is proportional to the relative change of the causing stimulus.

$$dp = -k \frac{dS}{S} \quad (1)$$

where dp is the differential change in perceived intensity, dS is the differential increase in the stimulus' intensity, S is the stimulus' intensity at the instant and k is a positive constant determined by the nature of the stimulus. However, stimuli whose growths produce decreasing perception intensity, e.g. distance, dissimilarity, discontinuity that are used in the proposed algorithm, can be described by assuming that the proportionality constant is negative. Integration of the last equation results in

$$p = -k \ln S + C \quad (2)$$

where C is the constant of integration. Assuming zero perceived intensity, the value of C can be found as

$$C = k \ln S_0 \quad (3)$$

where S_0 is the stimulus' value that results in zero perception and under which no stimulus' change is noticeable. Combining the above formulas it can be derived that

$$p = -k \ln \frac{S}{S_0} \quad (4)$$

Figure 1 presents the response obtained by such a function.

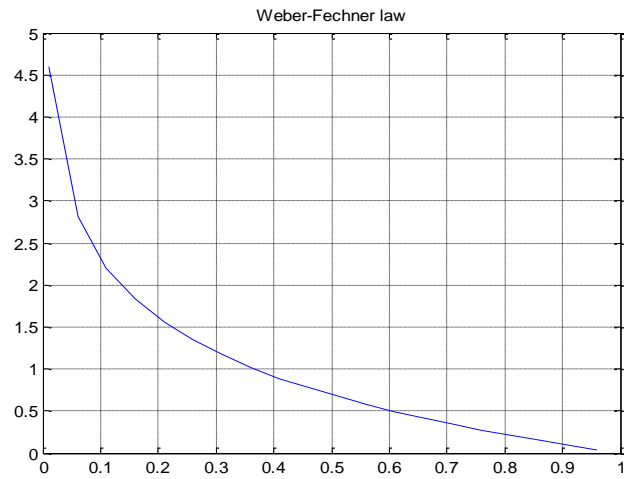


Figure 1 Perceived intensity response according to the Weber-Fechner law

Appendix E:

Powell's Method

Powell's method, strictly **Powell's conjugate gradient descent method**, is an algorithm proposed by Michael J. D. Powell for finding a local minimum of a function. The function need not be differentiable, and no derivatives are taken.

The function must be a real-valued function of a fixed number of real-valued inputs, creating an N -dimensional hypersurface or Hamiltonian. The caller passes in the initial point. The caller also passes in a set of initial search vectors. Typically N search vectors are passed in which are simply the normals aligned to each axis.

The method minimizes the function by a bi-directional search along each search vector, in turn. The new position can then be expressed as a linear combination of the search vectors. The new displacement vector becomes a new search vector, and is added to the end of the search vector list. Meanwhile the search vector which contributed most to the new direction, i.e. the one which was most successful, is deleted from the search vector list. The algorithm iterates an arbitrary number of times until no significant improvement is made.

The method is useful for calculating the local minimum of a continuous but complex function, especially one without an underlying mathematical definition, because it is not necessary to take derivatives. The basic algorithm is simple, the complexity is in the linear searches along the search vectors, which can be achieved via Brent's method.

The essence of Powell's method is to add two steps to the process described in the preceding paragraph. The vector $\vec{P}_n - \vec{P}_0$ represents, in some sense, the average direction moved over

the n intermediate steps $\vec{P}_0, \vec{P}_1, \dots, \vec{P}_n$ in an iteration. Thus the point \vec{X}_1 is determined to be the point at which the minimum of the function f occurs along the vector $\vec{P}_n - \vec{P}_0$. As before, f is a function of one variable along this vector and the minimization could be accomplished with an application of the golden ratio or Fibonacci searches. Finally, since the vector $\vec{P}_n - \vec{P}_0$ was such a good direction, it replaces one of the direction vectors for the next iteration. The iteration is then repeated using the new set of direction vectors to generate a sequence of points $\{\vec{X}_k\}_{k=0}^{\infty}$. In one step of the iteration instead of a zig-zag path the iteration follows a "dog-leg" path. The process is outlined below.

Let \vec{X}_0 be an initial guess at the location of the minimum of the function

$$z = f(\vec{X}) = f(X_1, X_2, \dots, X_n)$$

Let $\vec{E}_k = (0, \dots, l_k, 0, \dots, 0)$ for $k = 1, 2, \dots, n$ be the set of standard base vectors.

Initialize the vectors $\vec{U}_k = \vec{E}_k$ for $k = 1, 2, \dots, n$ and use their transpose \vec{U}'_k to form the columns of the matrix U as follows:

$$U = [\vec{U}'_1, \vec{U}'_2, \dots, \vec{U}'_n]$$

Initialize the counter $i = 0$

(i) Set $\vec{P}_0 = \vec{X}_i$.

(ii) For $k = 1, 2, \dots, n$.

find the value of $\gamma = \gamma_k$ that minimizes $f(\vec{P}_{k-1} + \gamma \vec{U}_k)$, and set $\vec{P}_k = \vec{P}_{k-1} + \gamma_k \vec{U}_k$.

(iii) Set $\vec{U}_j = \vec{U}_{j+1}$ for $j = 1, 2, \dots, n-1$ and set $\vec{U}_n = \vec{P}_n - \vec{P}_0$

(iv) Increment the counter $i = i + 1$.

- (v) Find the value of $\gamma = \gamma_{min}$ that minimizes $f(\vec{P}_0 + \gamma \vec{U}_n)$, and set $\vec{X}_i = \vec{P}_0 + \gamma_{min} \vec{U}_n$.
- (vi) Repeat steps (i) through (v) until convergence is achieved.

A typical sequence of points generated by Powell's method is shown in Figure 2 below.

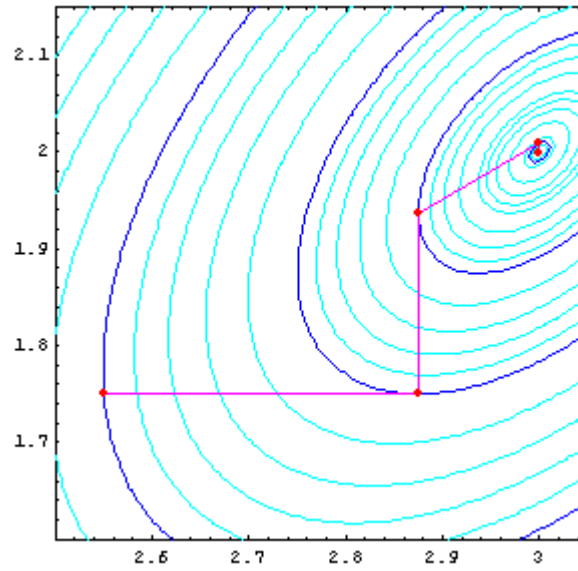


Figure 1. A sequence of points in 2D generated by Powell's method.

List of publications

Journal papers:

- [1] **Daolei Wang**, Kah Bin Lim, "Obtaining depth map form segment-based stereo matching using graph cuts", *Journal of Visual Communication and Image Representation* 22, pp. 325-331, 2012.
- [2] Kah Bin Lim, **Daolei Wang**, Wei loon Kee, "Virtual cameras rectification with geometrical approach on single-lens stereovision using a biprism", *Journal of Electronic Imaging*, 21(2), 023003, 2012.
- [3] **Daolei Wang**, Kah Bin Lim, "Geometrical Approach for Rectification on Single-Lens Stereovision Using a Triprism", *Machine Vision and Applications*, accepted, 2012.
- [4] Wei loon Kee, Kah Bin Lim, **Daolei Wang**, "Virtual Epipolar Line Construction of Single-Lens Bi-prism Stereovision System", *Journal of electronic science and technology*, vol. 10, No. 2, June 2012.
- [5] Xiaoyu Cui, Kah Bin Lim, Qiyong Guo, **Daolei Wang** "An accurate geometrical optics model for single-lens stereovision system using prism", *The Journal of the Optical Society of America A (JOSA A)*, Vol. 29, Issue 9, pp. 1828-1837, 2012.
- [6] Kah Bin Lim, **Daolei Wang**, Wei loon Kee, "3D scene reconstruction based on single-lens stereovision system using a bi-prism", *Journal of Computer Animation and Virtual Worlds*, submitted, 2012.
- [7] Wei loon Kee, Kah Bin Lim, **Daolei Wang**, "Solving Stereo Correspondence Problem of Single-Lens Bi-prism Stereovision System Using Geometrical Approach", submitted, 2012.

Conference papers:

- [8] **Daolei Wang**, Kah Bin Lim , "A new segment-based stereo matching using graph cuts," *Computer Science and Information Technology (ICCSIT)*, 2010 3rd IEEE International Conference on , vol.5, no., pp.410-416, 9-11 July 2010.